

中华学习机使用入门

卓小越 编著

電子工業出版社

中华学习机使用入门

卓小越　编著

電子工業出版社

内 容 简 介

中华学习机是青少年学习计算机和中小学进行计算机辅助教学的有力工具，也是计算机爱好者的优选机型。本书介绍中华学习机的基本连接装配方法、操作步骤、使用技巧、出现的一些问题的解决办法、编辑汇编程序的使用以及一些实用有趣的程序。读了本书，必将为您学习计算机铺上一条捷径，给您学习计算机带来莫大兴趣。

中华学习机使用入门
卓小越 编著
责任编辑 路石
·
电子工业出版社出版（北京海淀区万寿路）
电子工业出版社发行 各地新华书店经销
妙峰山印刷厂印刷
·
开本：787×1092毫米1/32 印张：4.375 字数：97千字
1989年9月第一版 1990年5月第二次印刷
印数：8000—28100册 定价：1.90元
ISBN 7-5053-0627-8/TP·99

目　　录

第一章　概况和接线……………………………………1
1.1　概况……………………………………………1
1.2　基本系统的接线……………………………………2
1.3　加电启动进入BASIC……………………………5
1.4　介绍一个游戏程序“打伞兵”……………………6
第二章　基本操作……………………………………7
2.1　键盘介绍……………………………………………7
2.2　自检……………………………………………10
2.3　系统复位……………………………………………12
2.4　屏幕编辑命令……………………………………12
2.5　从磁带装入一个音乐程序……………………………15
第三章　汉字系统的使用……………………………17
3.1　进入和退出汉字系统……………………………17
3.2　输入方式的切换……………………………………18
3.3　状态字符的显示与关闭……………………………18
3.4　拼音输入法……………………………………19
3.5　特殊符号的输入……………………………………20
3.6　区位输入法……………………………………20
3.7　用拼音输入法写一首诗……………………………21
第四章　磁带机的使用……………………………22
4.1　磁带文件的存取……………………………………22
4.2　游戏程序的装入执行……………………………23
4.3　设计一个音乐程序……………………………………24

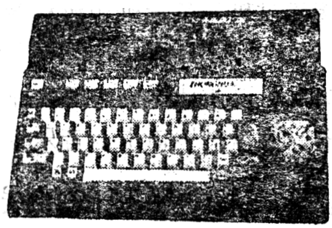
第五章　磁盘机的连接和DOS的使用……………………26
5.1　磁盘机的连接……………………26
5.2　磁盘机加电启动的过程……………………27
5.3　DOS的使用……………………27
5.4　使用两个磁盘机进行文件拷贝……………………34
5.5　介绍一个游戏程序“击剑”……………………35
第六章　打印机的连接和使用……………………36
6.1　扩充槽的使用……………………36
6.2　打印机的连接……………………36
6.3　西文方式的打印……………………36
6.4　汉字方式的打印……………………37
6.5　编写图形中有汉字的程序……………………39
6.6　屏幕硬拷贝……………………40
6.7　介绍一个游戏程序“中国象棋”……………………42
第七章　LOGO语言……………………44
7.1　LOGO语言的介绍……………………44
7.2　LOGO语言的进入和退出……………………44
7.3　LOGO语言的画图功能……………………45
7.4　LOGO语言的运行方式……………………47
7.5　LOGO语言的其他功能……………………48
第八章　监控程序……………………49
8.1　监控程序的功能……………………49
8.2　监控命令的使用……………………49
8.3　小汇编的使用……………………59
第九章　中华学习机用法讨论……………………61
9.1　LOGO语言……………………61
9.2　汉字方式下的彩色显示……………………61
9.3　录音机的接线和使用……………………62
9.4　中断的使用……………………63

9.5 CEC-I 1.1 版软件中的若干问题及解决方法 ………65
第十章 编辑汇编程序的使用 ……………………………………66
10.1 汇编语言程序概述 ……………………………………66
10.2 汇编过程简介 …………………………………………68
10.3 6502编辑/汇编程序的进入和退出 …………………71
10.4 编辑/汇编提供的命令 ………………………………73
10.5 源文件的建立 …………………………………………75
10.6 汇编命令和汇编过程的控制 …………………………87
10.7 源程序的编辑 …………………………………………89
附录……………………………………………………………102
附录A 助记符指令表，状态寄存器…………………………102
附录B 带有操作码、执行周期和存储要求的指令表………104
附录C 字母、字符显示码表（基本字符组）………………127
附录D 区位码、国标码、内码对照表………………………129

第一章　概况和接线

1.1　概况

CEC-Ⅰ型中华学习机是一种灵巧型计算机。它适合于中小学及家庭使用，并且将成为广大青少年学习计算机和进行计算机辅助教学的有力工具。



中华学习机的主机和键盘是制作在一起的。它的所有电路都制作在主机内的一块电路板上。整个学习机长37厘米、宽27厘米、高8厘米，重量为2.43千克，便于安置，便于携带。如右图所示。

中华学习机的结构和工作方式比较灵活，它的汉字功能部件和磁盘机接口部件是可以选择的，用户可以根据需要取舍。主机配上家用彩色电视机或黑白电视机以及盒式录音机就可以组成基本系统。目前，一般家庭都有电视机和录音机，因此，只要有了主机就可以在家里用上计算机了。如果再配上汉字功能部件以及磁盘机接口部件，接上磁盘机，就可以构成一台功能较强而且有汉字处理功能的微型计算机系统。

CEC-Ⅰ型中华学习机与目前世界上比较流行的苹果Ⅱ微型机的功能相当，并且还有些增加，它可以运行苹果Ⅱ机上运行的上万种软件，包括科学计算软件、中小学辅助教学

软件以及游戏软件等。中华学习机的主机上有监控程序、BASIC语言、LOGO语言以及汉字软件。因此，一开机就可以使用这些语言而不需要从磁盘或磁带上读入计算机。

CEC-Ⅰ型中华学习机具有汉字处理能力。可以使用拼音方法将汉字输入计算机。主机内可配上汉字字库，字库中提供国家标准规定的6763个汉字，还有日文、英文、俄文字母等。显示汉字时，满屏幕可显示170个汉字。打印汉字时，可以在打印机上打印15种字形。

CEC-Ⅰ型中华学习机的主机电路板上装有磁盘机、录音机和游戏操作杆的接口电路，有了这些接口电路就可以连接磁盘机、录音机和游戏操纵杆。

CEC-Ⅰ型中华学习机留有扩充槽口。通过这个扩充槽口，可以使计算机连接各种各样的扩充卡。例如打印机接口卡、磁盘机接口卡、音乐卡等等。

1.2 基本系统的接线

1.2.1 电源线

要使用中华学习机首先要连接电源线，就像家里的电视机一样，没有电什么电视节目也看不到。那么电源线在哪里呢？如果你面对机壳背后，则电源线是右端引出的一根固定引线，它不能从机壳上拔下来，参看图1-1。这根电源引线的末端是一个插头，它可以插在220伏电源的插座上，也就是一般家用电源的插座。在插入插座之前，先检查一下电源开关。如果面对键盘，电源开关在机壳左面。如果开关是在开的位置，即船形开关标有白点的一侧被按进去了，则应把它拨到关的位置，即将船形开关没有白点的一侧按进去。然后将电源插头插入220伏电源插座上，这样电源线就连接好

了。

1.2.2 显示器的连接

电源线接好之后，可以连接显示器。中华学习机的显示器可以使用黑白电视机或彩色电视机；也可以使用单色监视器或彩色监视器。对于彩色监视器，应该使用PAL制式的，也就是我国电视台所采用的制式。

1. 电视机的连接

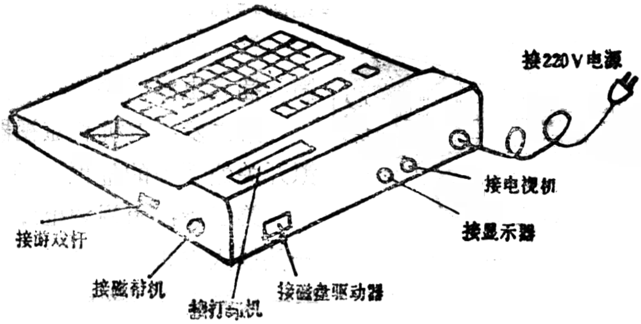


图 1-1 接口位置

当你面对机壳背后时，从右向左有两个插孔。右边的插孔标有电视机字样，可以连接电视机，参见图1-1。连线方法如下：

第一步：把电视机的天线插头去掉，音量开关拧到最小。

第二步：将电视机电源线插头插入电源插座。

第三步：主机和电视机电源开关处于关的位置。

第四步：用视频电缆线的

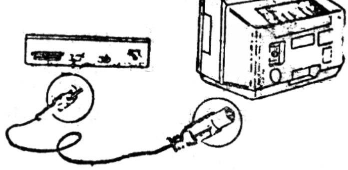


图 1-2 电视机的连线

一头插入主机的电视机插孔，另一头插入电视机天线插孔。电视机天线插孔的输入阻抗为75Ω。参看图1-2。

如果你的电视机有视频输入端，你可以将主机的监视器输出信号接到视频输入端上。这样可以使屏幕上的画面更加清晰稳定。

2. 监视器的连接

如果面对主机背后，从右边向左数第二个插孔标有监视器字样，可以连接监视器。参见图1-1。连接方法如下：

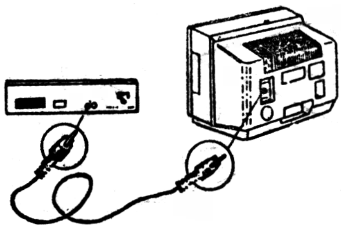


图 1-3 监视器的连线

第一步：将监视器电源线插头插入电源插座。

第二步：主机和监视器电源开关处于关的位置。

第三步：用视频电缆线的一头插入主机的监视器插孔，另一头插入监视器标有IN的插孔。参见图1-3。

当电源线和显示器线连好后，一个最小系统就安装好了。在这个最小系统上，可以作自检测试，可以敲入执行BASIC程序，但是不能保存程序，因此还要安装磁带机或磁盘机来保存程序。

1.2.3 磁带机的连接

中华学习机可以使用普通盒式录音机作为外部存贮器来保存程序。当然，如果有专配微型机的盒式磁带机更好。在接线之前，首先要找到主机上磁带机的插座位置。在机壳的右侧有两个插座，右边一个圆型插座是磁带机插座。参见图1-1。

接线步骤如下：

第一步：关掉主机电源。

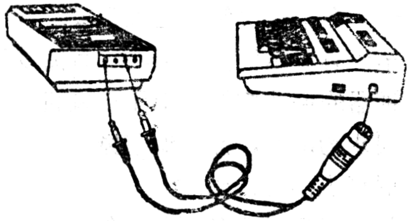


图 1-4 录音机的连线

第二步：把录音机电缆线一端的五芯插头插入主机上的磁带机插座。注意，五芯插头的凹口部分向着正上方。

第三步：将录音机电缆线另一端的两个插头分别插入录音机的插孔。注意，引线上标有MIC或OUT的小插头插入录音机的MIC插孔，引线上标有EAR或IN的小插头插入录音机的EAR插孔（或按厂家说明）。

第四步：将录音机电源线插头插到电源插座上。参见图1-4。

1.3 加电启动进入BASIC

当基本系统的连线接好之后，就可以加电了。先开显示器电源开关，即电视机开关或监视器开关，然后再开学习机电源开关。

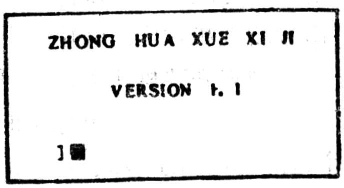


图 5-1

开机后，机壳右上方的指示灯变亮，注意指示灯在右上方的小孔内。显示器上显示中华学习机的汉语拼音和版本号，如图1-5所示。屏幕左下角出现BASIC语言提示符“]”和闪动的光标。这表示已进入BASIC语言，等待输入BASIC命令。这时，可以使用BASIC来编写程序了。

在第一次使用电视机或监视器时，有可能屏幕不清楚，甚至没有显示任何信息，这时需要调整电视机的频道选择或监视器的调节旋钮。对于电视机应将频道放在二频道（以厂家要求为准），然后调节频道微调及对比度、亮度旋钮，直到屏幕上出现清晰的字符。如果是彩色电视机，还可以通过自检方式显示彩条图象，然后调节频道微调及色饱和度、亮度、对比度旋钮、直到屏幕上的彩条满意为止。以后还可以把此频道作为计算机专用频道，再开机就不用调机了。对于监视器应该调节监视器上的同步调节旋钮，使图像清晰稳定。

在关掉主机电源之后不要立即开电源。如果需要开电源，应该等15秒钟以后再开。

1.4 介绍一个游戏程序“打伞兵”

作为本章的结束，我们来装入执行一个游戏程序，这个游戏程序的名字叫“打伞兵”。游戏的玩法是这样的：屏幕上有一个炮台，炮台上的大炮可以由人来控制，通过敲F键和D键来改变大炮炮口的方向，通过敲其他键来开炮。屏幕上方有飞机不断开过，飞机上不断有伞兵跳下来。玩游戏的人要尽量通过控制大炮将伞兵打中或将飞机打中。打中一个伞兵可得2分，分数写在炮台上，打中直升飞机或轰炸机可得5分，打中一颗轰炸机扔下的炸弹可得25分。如果你不能全部消灭伞兵，让他们之中的4个人落在炮台左边或右边，他们就会搭人梯炸毁你的大炮，同样，如果你不打掉飞机扔下的炸弹，你的大炮也会被炸毁，所以你要争取在大炮被敌人毁掉之前，尽量多得分。不过还要注意节约炮弹，因为用掉一颗炮弹就要减一分。

第二章　基本操作

2.1　键盘介绍

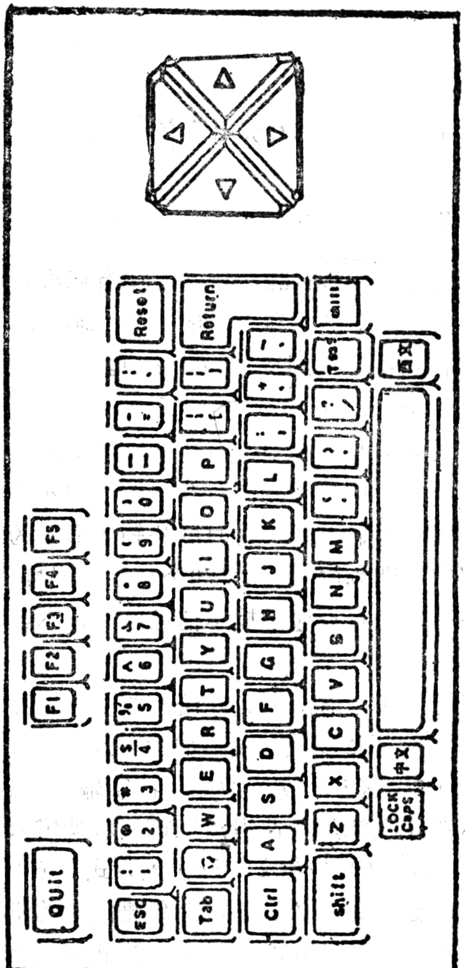
中华学习机的键盘一共有69个键。其中有标准打字机键盘的各种键。例如，数字键 0 到 9，字母键A到Z等。除标准键外，还有一些特殊键。键的排列如图 2－1。

这些键的作用介绍如下：

“中文”键的功能是进入中文方式。当按下此键后，系统进入中文方式，显示屏幕变为高分辨率图形显示方式，使用高分辨率第二页作为显示区。这时屏幕输出显示为满屏10行，每行17个汉字，即满屏可显示170个汉字。屏幕最底行即第11行为状态行，此行只显示提示信息，不显示屏幕输出信息。

“西文”键的功能是退出中文方式。当按下此键后，系统退出中文方式，显示屏幕回到文本显示方式。

“F1”～“F5”这五个功能键用于汉字输入方式的切换。也就是进入汉字系统后，使用这些键来改变输入方式。“F1”是字母方式，也就是英文字母输入方式，或叫ASCII字符输入方式，在这种输入方式下，系统把敲入的键码解释为英文字母。“F2”是拼音方式，在这种方式下，系统把敲入的键解释为汉语拼音。“F3”是区位方式，在这种方式下，系统把敲入的键解释为国标的区位码，当然此时只能敲入数字键。“F4”和“F5”暂时没有使用，它们留作扩充输



Quit
F1
F2
F3
F4
F5
Esc
Tab
Ctrl
shift
Reset
Return
LOCK Caps
中文
西文

图 2-1

入方式时使用，在汉字系统的介绍中还会详细说明。

“Quit”键用于从正在运行的BASIC程序中退出，回到提示状态。例如，当执行下面程序后，屏幕总是不断显示“ABC”。

```
10  pRINT  "ABC"
20  COTO  10
RUN
```

当按“Quit”键后退出程序，回到提示状态。

“Test”键是用于测试计算机的存贮器和显示器。当同时按下“Ctrl”键、“Reset”键和“Test”键后，机器就进入检测状态，有了这个功能我们就可以检查计算机的存贮器是否有故障。

“Shift”键用于产生上档键码。例如等于号“=”和加号“+”是在一个键上，直接按这个键，输入的是“=”号，按下“Shift”再按这个键，输入的是“+”号。

“Ctrl”键与其他键合用可以产生另外一套键码。例如，单独按下A键输入计算机的是A的键码。而按下“Ctrl”键同时按下A键输入计算机的是另外一个键码。单独按“Ctrl”键不起任何作用。因此在放开两个键时，应同时放开或先放字母键后放“Ctrl”键。

“Reset”键用于系统复位。按下“Ctrl”键同时按下“Reset”键就产生系统复位。在本节后面还会讲到系统复位的作用。

“Esc”键和◁，▷，▽，△键用于屏幕编辑，即如果你敲错了键可以用这些键进行修改，本节后面还要详细介绍屏幕编辑命令。

“Return”键称为回车键，用于结束一个命令的输入，

也就是每个命令敲完后都要敲一下回车键，告诉计算机命令敲完了。有时，也用符号“↲”或“CR”表示回车键。

“Cap Lock”键用于锁定大写。当该键按下后，所有字母键的输入都为大写。当该键抬起时，字母键输入都为小写，此时按下Shift键同时再按字母键又可以得到大写。BASIC语言只认识大写的命令和语句，因此进入BASIC后要按下“Cap Lcck”键。

2.2 自检

当需要检查计算机的存贮器或者要进行彩色显示测试时，可以使用自检。

同时按下“Ctrl”键、“Reset”键和“Test”键，然后放开，机器就进入自检状态。自检程序首先检查RAM，即随机存贮器，随机存贮器里的内容在关掉电源后会消失。然后再检查ROM，即只读存贮器，只读存贮器里的内容在关掉电源后不会丢失。因此只读存贮器中的程序为固化的程序，或叫固化软件。在测试存贮器时，显示器上显示相应的检测内容和结果，如图2-2。

显示信息的意义如下：

左边一列是存贮器的名字，中间一列是存贮器的地址范围，右边一列是存贮器测试结果。如屏幕上显示的“RAM表示随机存贮器，地址从＄0000单元到＄BFFF单元共48K单元。其中＄0000和＄BFFF表示16进制数。48K单元就是48×1024个单元。“OK”表示都是好的。如有错，显示ERR。BNK1和BNK2是随机存贮器，地址从＄D000到＄FFFF。“ROM1”是只读存贮器，地址从＄80000到＄BFFF，内容是LOGO语言。“ROM2”是只读存贮器，地址是从＄D000到

```
MEMORY  —  TEST
           TIMES: 0001
- - - - - - - - - - - - - - - - - - - -
RAM     BFFF    OK
BNK1    FFFF    OK
BNK2    FFFF    OK
ROM1    BFFF    LOGO
ROM2    FFFF    CEC-BASIC
AUX1    BFFF    HZTABLE
AUX2    FFFF    HZPROGRAM
AUX3    7FFF    CECWL
```

图 2-2

$FFFF，内容是CEC-BASIC和监控程序。“AUX1”是只读存贮器，地址是从$8000到$BFFF，主要内容是汉字码表内容。“AUX2”是只读存贮器，地址是从$D000到$FFFF，主要内容是汉字管理系统。“AUX3”是汉字字库。

测试正确时，屏幕显示相应的测试内容的名称。如果测试出不为系统所知的内容，屏幕在相应位置显示UNKNOWN。当没有汉字库时，屏幕在相应位置显示NULL。

当存贮器测试完后，约等15秒，进入彩色显示测试，屏幕将显示16种不同颜色的彩条。对于黑白电视机或单色监视器，则显示不同精度的线条。

自检程序可以连续循环地测试，并在TIMES：一栏显示测试的次数。如需调试彩色电视机，可在屏幕显示彩条时按任何一键，使屏幕一直显示彩条，直到调试完彩色电视机，再按任一键又进入自检程序循环测试。

如果要退出自检，可以使用系统复位，即同时按下Ctrl和

Reset键，计算机便进入BASIC状态。

2.3 系统复位

当加电后系统运行正常。但由于操作失误或程序错误等原因使系统进入死循环，此时敲什么键都没用；或者使用者需要系统重新启动工作时，如从自检状态退出，可以同时按下Ctrl键和Reset键，这样系统便重新启动进入BASIC。我们把这个过程叫系统复位。

2.4 屏幕编辑命令

当我们用键盘输入BASIC语言或命令时，免不了会敲错键，或者对输入的语句需要修改。中华学习机提供了一套屏幕编辑命令，用户可以使用这套屏幕编辑命令来修改已经输入或正在输入的BASIC程序。下面我们逐个介绍这些屏幕编辑功能。

2.4.1 ◁键

当我们按下一个字符键时，这个键的键码就送入内存中的键盘输入行。同时，这个键的字符显示到当前屏幕的光标位置上，光标向右进一格。当按下◁键时，删去键盘输入行中最末一个字符，也就是最后输入的那个键码，并使屏幕上光标向左退一格，但屏幕上显示的字符还留在屏幕上。当我们要删除输入的字符时，可以使用这个键。例如，我们开始想敲LET却敲成了LRT，此时，敲两下◁键，使光标在R字上闪动，然后敲E键就可以把R改变成E。

2.4.2 ▷键

按下▷键，则光标处的字符输入到内存中的键盘输入行

的末尾，并使光标右移一格。例如 我们要敲入

10 PRINT "ABCDEF"

但敲成了

10 PEINT "ABCDRF"

这时，可以用◁键把光标退到"R"上，敲入正确的字符E后，用▽键使光标移过E右边的每个字符，这些字符就输入到内存中的键盘输入行了。

2.4.3 △键和▽键

按下△键，光标上移一行，不改变键盘输入行的内容，按下▽键，光标下移一行，不改变键盘输入行的内容。

2.4.4 Ctrl-X

按下Ctrl键同时按下X键，删去内存中当前输入行。屏幕上显示："＼"字符，并回车换行，给出新的提示符"]"，等待输入新的一行。例如，敲入了下面的内容

]ABCDEFG

然后同时敲Ctrl和X，屏幕显示为

]ABCDEFG＼

]

.4.5 "ESC"键

除上述编辑命令外，系统还提供一套只改变屏幕上显示的字符和光标位置而不改变键盘输入行字符的编辑命令，这些编辑命令都是以敲入"ESC"键开始的。敲了"ESC"键后就进入编辑状态，在此状态下，有11个键（A，B，C，D，E，F，I，J，K，M @）具有编辑功能，称为"ESCAPE码"或叫编辑码。在执行一个编辑码后是否退出了编辑状态取决于执行的是哪一种编辑码。如果执行的是A，B，C，D，E，F，@这7个码，则执行后自动退出编辑状态。

如果执行的是I，J，K，M这4个码，则执行后仍处于编辑状态，直到敲入一个非编辑码才退出编辑状态，也就是说，在敲了“ESC”键后，I，J，K，M这四个键可以连续敲，屏幕上的光标连续移动。这些键的功能规定如下：

“ESC”A：光标右移一格

“ESC”B：光标左移一格

“ESC”C：光标下移一行

“ESC”D：光标上移一行

“ESC”E：从光标清除到行尾。例如已敲入下述信息：

］ABCDEFGHIJKLMNOPQRSTUVWXYZ

光标在G字符上敲入“ESC”E后，屏幕显示为

］ABCDEF

“ESC”F：从光标清除到屏幕尾。例如，屏幕显示为：

```
]10  PRINT "ABC"
 20  PRINT "EFG"
 30  PRINT "HIJKM"
```

光标在语句10行的A字符上，敲入“ESC”F后，屏幕上显示为

```
]10  PRINT "
```

“ESC”@：清屏幕，光标回到左上角。

“ESC”I：光标上移一行，仍处于编辑状态。

“ESC”J：光标左移一格，仍处于编辑状态。

“ESC”K：光标右移一格，仍处于编辑状态。

“ESC”M：光标下移一行，仍处于编辑状态。

例如，要修改一个程序，可先用LIST列出程序，再用“ESC”和I，J，K，M键把光标移到要修改的程序行首，用▷键把光标移到出错的字符上，修改后，再用▷键把后面

的字符输入计算机，最后敲回车键。

除上述以键盘方式提供的屏幕编辑命令外，在汉字显示方式下，系统还提供一组以程序方式对屏幕进行编辑的命令和一些输出控制字符。即：

CHR$(7)　扬声器“嘟”叫一声

CHR$(8)　光标退一格

CHR$(10)　光标下移一行

CHR$(11)　从光标清至屏幕尾

CHR$(12)　清屏幕，光标移到左上角

CHR$(13)　输出回车符，光标移至下一行

CHR$(14)　置显示方式为正常方式

CHR$(15)　置显示方式为反相方式

CHR$(19)　暂停输出，等待按任一键恢复输出

CHR$(26)　清除光标所在行

CHR$(29)　从光标清至行尾

假如我们用N来表示括号中的数，上述功能的实现应该用下述语句实现，即：

```
10  PRINT  CHR$(N)
```

例如敲入语句PRINT　CHR$(7)，然后回车，计算机就会“嘟”地叫一声。若放在程序中只需加上行号即可。

2.5　从磁带装入一个音乐程序

作为本章的结尾，我们从磁带装入并执行一个音乐程序，这个程序演奏的曲子是中华人民共和国国歌，怎样在学习机上编写音乐程序将在第四节磁带机的使用中介绍。下面是演奏国歌的程序：

```
100  READ  X, Y
```

```
110 IF X=0 THEN END
120 MUSIC X, Y
130 GOTO 100
140 DATA 255, 70, 192, 160, 192, 70, 192,
    70, 192, 70, 255, 70, 228, 30, 204, 30, 192,
    160, 192, 160, 1, 70, 152, 70, 192, 70, 172,
    30, 152, 30, 128, 160, 128, 160, 152, 110,
    152, 30, 192, 110, 152, 30
150 DATA 128, 110, 152, 30, 172, 160, 172,
    160, 172, 160, 114, 160, 128, 160, 172, 160,
    152, 160, 128, 70, 152, 160, 128, 70, 152,
    70, 172, 30, 152, 30, 192, 160
160 DATA 152, 160, 152, 160, 255, 70, 228,
    70, 192, 70, 192, 70, 152, 110, 152,30, 128,
    70, 128, 70, 172, 70, 172, 30, 172, 30, 114,
    160, 172, 160, 172, 70, 255, 70, 192, 160,
    192, 70, 192, 70
170 DATA 152, 160, 152, 70, 152, 70, 128,
    160, 128, 160, 128, 160, 128, 160, 192, 70,
    152, 70, 128, 70, 128, 30, 128, 30, 114,
    160, 128, 160, 152, 110, 192, 30, 128, 70,
    128, 30, 128, 30
180 DATA 152, 160, 192, 160, 255, 160, 192,
    160, 152, 110, 192, 30, 128, 70, 128, 30,
    128, 30, 152, 160, 192, 160, 255, 160, 192,
    160, 255, 160, 192, 160
190 DATA 255, 160, 192, 160, 192, 160, 0, 0
```

第三章　汉字系统的使用

3.1　进入和退出汉字系统

进入和退出汉字系统有两种方式。一种是键盘方式，即通过敲键进入汉字系统。另一种是程序方式，即在程序执行时才进入汉字系统。我们首先介绍键盘方式。

3.1.1　键盘方式

当加电后屏幕显示方式为字符方式，也就是说屏幕此时只能显示字符。当按下中文键后，系统进入汉字方式。此时屏幕显示方式变为图形方式，也就是说，此时的汉字和字符是以图形方式显示出来的，这时屏幕每行可显示17个汉字或34个字符，共10行，也就是共170个汉字或340个字符。屏幕底部第11行为状态行，状态行左端显示“字母”两字，表示系统当前处于字符输入状态。在字符输入状态下，敲键产生的是英文字母。

当需要退出汉字方式时，可按西文键，屏幕回到字符显示方式。

3.1.2　程序方式

除键盘方式进入汉字系统外，程序方式也可以进入汉字系统。在BASIC程序中使用下述语句可以进入汉字系统：

```
PR#3：PRINT
```

例如：

```
]10　PR#3：PRINT
```

执行该语句后就进入汉字系统。

使用TEXT语句可以退出汉字系统，例如：

```
] 20  TEXT
] RUN
```

回车后退出汉字系统。

3.2 输入方式的切换

在进入汉字系统后，状态行左端显示“字母”两字，表示系统处于字符输入状态，在此状态下，敲键产生的是英文字母。

当按下F2键时，状态行左端显示“拼音”两字，表示系统处于汉字的拼音输入方式。在此方式下，敲键产生的是拼音字母和拼出的汉字。

当按下F3键时，状态行左端显示“区位”两字，表示系统处于汉字的区位方式 在此方式下，只能敲数字键，表示区位码，并产生区位码对应的汉字，区位码在后面还会详细介绍。

当按下F1键时，又回到“字母”方式。

F4，F5键用于扩充汉字方式。用户程序可以修改存贮器中的指针值，使键盘输入由用户程序处理。详细的修改方法可参考中华学习机技术参考手册（软件）。

3.3 状态字符的显示与关闭

同时按下Ctrl键和O键，即Ctrl-O键，可控制状态字符的显示与不显示，但不改变输入方式。例如，刚进入汉字方式时，状态行左端有字母两字，按Ctrl-O键后字母两字就不见了。再按一次Ctrl-O键后字母就又出现了。在程序中可以用PRINT CHR＄(18)来清除状态提示。

3.4 拼音输入法

按下F2键后进入拼音输入方式。

中华学习机的拼音输入法采用全拼音输入，即按声母、韵母全部敲入。计算机根据输入的音节在状态行上提示出该音节的所有汉字，包括不同声调的汉字，操作者根据提示选择所需要的汉字。因此，只要学过汉语拼音的人就都会使用这种输入方法。

3.4.1 第一键提示常用字

当敲入第一键后，虽然只有声母，但屏幕状态行上显示出以该声母开头的音节中最常用的6个汉字。此时，若包含有要输入的汉字，只要敲入相应的序号即可，若没有，则要继续敲入韵母。例如敲入b后，状态行上提示出：

拼音：b 1 不 2 部 3 本 4 表 5 把 6 变

3.4.2 用V表示ü

按照汉语拼音的规定，ü的前面为Y，并独立发音时如YU、YUE，或者和J、Q、X相拼时都要去掉点写成U。因此，在中华学习机全拼音方案中，应该去点时仍敲U键，不去掉点时敲V键。

例如，要输入“于”，应该敲YU，要输入“旅”，应该敲LV。

4.3.3 寻找同音节字

当敲完韵母后，屏幕状态行提示出6个同音节字。若这6个同音节字中没有要输入的汉字，可敲句号键“>.”，每敲一下“>.”键，屏幕状态行就显示出下一幕6个同音节字，直到计算机发出“嘟”声，表示已全部显示完。

当要寻找前一幕汉字时，可敲逗号键“<”，此时前一幕6个汉字显示在状态行上。

3.4.4 敲入所选字的序号

当屏幕状态行上出现要输入的汉字时，只要敲入该汉字前面的序号，该汉字就显示在屏幕光标位置上，且汉字内码输入到键盘输入行。连续敲入序号，可连续输入汉字。

3.4.5 删除拼音提示

若输入时，有拼音敲错，可直接按◁键，使状态行上最后一个拼音字母被删除，或者按空格键使所有提示字符都删除。当状态行上的拼音都删掉后，再按◁键，则删除光标上的信息。例如要敲入“中国”两字，“中”已经输入，“国”的拼音为guo，假定敲成了gua，此时可以敲一下◁键，于是状态行上只剩gu，再敲o便得到“国”的提示。

3.5 特殊符号的输入

在拼音方式下，按下减号键可输入标点符号，如逗号、句号等。按下等号键可输入算术运算符号，如加号、减号等。按下反斜线键可以输入制表符号，如横线，竖线等。用句号键“>”和逗号键“<”可以寻找下一幕或前一幕特殊符号。

3.6 区位输入法

按下F3键后进入区位输入方式。区位输入方式就是通过敲入汉字的区码和位码来输入汉字，一个汉字对应哪个区码和哪个位码要查找国家规定的区位码表才能知道。

区位码可输入的内容很多，包括常用汉字6763个，各种符号202个（如间隔符号、标点符号、运算符号、制表符号、

单位符号等），序号60个（包括10～20、（1）～(20)、（一）～（十）），数字22个（包括0～9，Ⅰ～Ⅻ），英文字母大小写52个，日文假名169个，希腊字母大小写共48个，俄文字母大小写共66个。实际上，就是汉字字库中的所有字符和汉字都可以用区位方式输入。

把所有上述字母、符号、汉字等划分成94个区，每个区有94个汉字或符号。因此每个汉字就有对应的区码和位码。而且区码取值在01到94之间，位码也在01到94之间。目前第10区到第15区以及第88区到第94区还没有规定内容。汉字是从第16区开始排放，一直到87区。

例如16区第1个字是“啊”，因此要输入“啊”字应该敲1601。当第四个键敲完后，“啊”字就立即出现在光标位置上。

当敲错区位码后，可按◁键，使区位码提示的最后一个数字被删除。若已敲入四个码，又敲◁键，则即删除第四个数字，也删除内存中键盘输入行中输入的汉字码，同时光标左移一格。

当区位码都删掉后，再按◁键，则删除光标位置上的信息。

3.7 用拼音输入法写一首诗

作为本章结尾，我们练习用拼音输入法敲入一首诗，并用BASIC语言把这首诗显示到屏幕上。

试敲入下述内容：

```
10  PRINT  “白日依山尽”
20  PRINT  “黄河入海流”
30  PRINT  “欲穷千里目”
40  PRINT  “更上一层楼”
```

然后，敲入RUN回车，于是这首诗便显示在屏幕上。

第四章　磁带机的使用

4.1　磁带文件的存取

为了把敲入计算机的程序保存起来，我们可以使用磁带和磁盘。这样，下次使用时就不用再敲一遍程序，而只要给计算机发一个命令，计算机就会从磁带或磁盘上把程序装入内存。

4.1.1　保存文件

在第一章中已介绍了磁带机的连线，即将磁带机连线一端的五芯插头插入主机右侧的圆形插座，另一端有两个插头，标有MIC或OUT的插头插入磁带机的MIC插孔，标有EAR或IN的插头插入磁带机的EAR插孔，然后将磁带放入录音机。

在BASIC的文件建好之后，敲入下述命令（注意文件名有引号）

] SAVE "文件名"

在按回车键之前，先按下录音机的REC和PLAY键，启动录音机录音，然后按下回车键，命令计算机把内存中文件存入磁带。当文件存好后，计算机显示提示符“]”，此时按下磁带机的STOP键。如果不敲入文件名，也可以将文件存入磁带，但要记住装入文件时磁带的初始位置，也就是录音机上的计数器值。因为在装入内存时，由于没有文件名就只能凭文件在磁带上的位置来装入文件了。

4.1.2 装入文件

当需要将磁带文件装入内存时，可以敲入

〕LOAD "文件名" （注意文件名有引号）

在按回车键之前，先按下录音机PLAY键，再按回车键。当文件装入后，屏幕显示被装入的文件名和提示符“〕”，表示可以输入新的命令，此时按下录音机的STOP键。

如果装入文件不成功，可能是录音机音量电位器的位置不合适。可以调节录音机音量旋钮，使录音机的工作电平处于一个合适的位置，然后重新装入一次，可能要试几次才能成功。成功后要记住音量旋钮的位置，以便以后再用时可以一次成功。

如果一盒磁带上装有多个文件，而需要装入的是磁带上较后面的文件，则计算机在装入第一个文件后显示出该文件名和一个反相显示的字符N，并继续装入后面的文件，直到装入LOAD命令所给出的文件名的文件为止。

也可以只敲入LOAD，不敲文件名。那么LOAD命令只将第一个文件调入内存，然后显示提示符“〕”。

4.2 游戏程序的装入执行

中华学习机配有各种磁带游戏程序，装入执行这些游戏程序的过程如下：

先将磁带退到开始位置，在键盘上敲入四个字母PLAY，然后按下录音机的PLAY键，使磁带机启动，再按下键盘上回车键。于是，过15秒钟左右，屏幕显示：

IT'S LOADING PLEASE WAIT…

当游戏程序装入内存后自动运行，此时按下录音机的

STOP键。若装入不成功，可以调节录音机的音量旋钮，再装一次，当音量旋钮调到一个适合的电平时，游戏程序便可装入执行。

4.3 设计一个音乐程序

中华学习机在BASIC程序中增加了音乐语句，可以用音乐语句较方便地编写简单的乐曲。

音乐语句的格式如下：

MUSIC X，Y

其中MUSIC表示音乐语句，它可以使计算机上的小喇叭发出声音。X表示发声的频率，也就是音阶。Y表示发声的时间，也就是拍子。X和Y的取值范围在0～255之间。它们的取值与对应的音阶和音长有一个对照表可以作为参考。此对照表如下所示，其中的取值可以根据用户的需要适当修改。

X值	255	228	204	192	172	152	144	128	114	102
音阶	$\underset{\cdot}{5}$	$\underset{\cdot}{6}$	$\underset{\cdot}{7}$	1	2	3	4	5	6	7

X值	96	84	75	71	62
音阶	$\dot{1}$	$\dot{2}$	$\dot{3}$	$\dot{4}$	$\dot{5}$

Y值	30	70	110	160	255
音长(拍)	$^1/_4$	$^1/_2$	1	$1\frac{1}{2}$	2

编写程序时可以用下面这种形式：

```
10  READ X, Y
20  IF X=0 GOTO 100
30  MUSIC X, Y
40  GOTO 10
```

```
50 DATA X, Y, X, Y, X, Y, ...
......
90 DATA 0, 0
100 END
```

程序的执行过程是这样的：先读X和Y，判断一下X中是否为0。若为零就结束程序的执行，若不为零就用MUSIC语句发出这个声音，然后再转到语句10去读下一组X和Y值，直到读到零值为止。从语句50开始放乐曲的谱子。所有音符放完后，放两个零作为结束。下面是乐曲“知了”的程序：

```
10 READ X, Y
20 IF X=0 GOTO 100
30 MUSIC X, Y
40 GOTO 10
50 DATA 152, 210, 128, 70, 114, 70, 95, 70,
   95, 70, 114, 70, 128, 140, 128, 70, 114, 70,
   128, 225, 128, 40
60 DATA 152, 210, 128, 70, 114, 70, 95, 70,
   114, 70, 128, 140, 128, 70, 114, 70, 128,
   255, 128, 40
70 DATA 75, 210, 84, 70, 75, 210, 84, 70,
   95, 210, 114, 70, 95, 210, 114, 70
80 DATA 128, 140, 75, 140, 84, 140, 114,
   140, 95, 255, 95, 40
90 DATA 0, 0
100 END
```

第五章　磁盘机的连接和 DOS的使用

磁盘机是一种高速的存贮设备。有了磁盘机就可以较快地把计算机内存中的程序存到磁盘上保存起来，下次开机后又可以较快地把程序从磁盘上装入计算机内存中。有了磁盘机我们的工作效率就可以提高很多倍。磁盘机分为硬磁盘机和软磁盘机，我们这里讲的是软磁盘机，也叫软盘驱动器。

5.1磁盘机的连接

在连接磁盘机之前先关掉主机电源。然后，找到主机上的磁盘机接口插座。磁盘机插座位于机盒背后，是一个20针扁平插头。将磁盘机电缆线一端的20线扁平插座与主机上的插头接好。注意，插针与插孔必须一一对应，电缆线插座上有凸起的部分朝上。电缆线另一端与磁盘机连接时，插座上

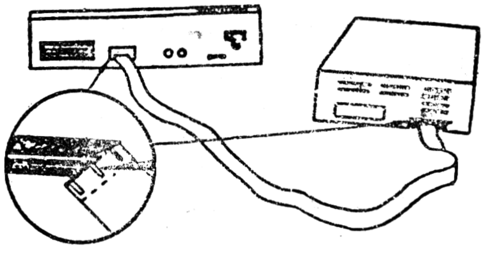


图　5-1　磁盘机的连接

有凸起的部分也朝上。参见图5-1。

5.2 磁盘机加电启动的过程

先将磁盘插入磁盘机，挟好磁盘（即关上小门），然后，接通显示器电源，最后接通主机电源。这时，系统会自动判断磁盘机中是否插入并挟好了软盘。若没插入软盘或没挟好软盘，系统都会直接进入BASIC，而不引导磁盘上的程序。只有当插入并挟好了软盘，系统才引导软盘上的程序。例如，若插入了有DOS(磁盘操作系统，下节详细介绍）的软盘，则系统自动引导DOS系统，引导结束后判断磁盘上有无HELLO文件，若有就把HELLO文件装入内存执行，若无就进入BASIC程序。

5.3 DOS的使用

DOS是磁盘操作系统，它的主要功能是管理磁盘和文件。DOS的操作命令很多，我们这里只介绍最常用的几个命令，其他命令的使用可参考中华学习机技术参考手册(软件)。

5.3.1 列磁盘文件目录

当开机装入DOS后，系统进入BASIC状态，敲入CATALOG↙可列出磁盘上的文件目录，列目录命令的一般格式为：

CATALOG, Ss, Dd

其中CATALOG是列目录命令，大写S表示槽口，小写s表示槽口号。大写D表示驱动器，小写d表示驱动器号，也就是磁盘机号。比如S6，D1表示6号槽口上的1号驱动器，S1，D2表示1号槽口上的2号驱动器。槽口号和驱动器号不用每次都敲入，DOS系统总是记住上次所访问过的槽口号

和驱动器号。如果只打入CATALOG，则DOS就列出上次访问过的磁盘驱动器上磁盘的文件目录。如果在开机后打入CATALOG，系统自动访问6号槽口1号驱动器。如下所示：

```
] CATALOG
DISK VOLUME 254
*A 008 HELLO
*I 018 ANIMALS
*T 003 APPLE PROMS
*I 006 APPLESOFT
*I 026 APPLEVISION
*B 017 BIORHYTHM
*B 008 MENU
*A 006 BRIAN'S THEME
*B 003 CHAIN
*I 009 COLOR DEMO
*A 009 COLOR DEMOSOFT
*I 009 COPY
*B 003 COPY .OBJO
*A 009 COPY A
*A 010 EXECDEMO
*B 020 FID
*B 050 FPBASIC
*B 050 INTBASIC
 B 004 CDS
*A 003 MAKE TEXT
*B 009 MASTER CREATE
*B 027 MUFFIN
```

```
*A  051  PHONELIST
*A  010  RANDOM
*A  013  RENUMBER
*A  039  RENUMBER INSTRUCTIONS
*A  003  RETRIEVE TEXT
*I  022  W.K
*I  000  ABC
*A  002  MINI
*B  003  MINI—ASSEMBLER
*B  010  BOOT13
```

从上面列出的内容我们可以得到这样一些信息：第一行是卷号254，一个软盘在初始化时给定一个卷号。左边第一列星号是文件的保护标志，它是操作系统提供的一项对文件写保护的措施。如果有星号，表示只能读这个文件不能改名或删除。如果作了改名或删除操作，DOS会告诉你文件有写保护，操作失败。如果没有星号则可以作各种操作。

第二列表示文件类型。A表示是BASIC程序，可以由BASIC系统直接解释执行。B表示二进制代码文件，这类文件不能由BASIC系统解释执行。二进制代码文件可以分成程序和数据两类。如果这个二进制代码文件是程序，则可以直接由计算机的中央处理器CPU来执行。如果这个二进制代码文件不是程序而是数据，则它不能被执行。T表示是文本文件，一般来说文本文件是不能直接执行的，但可以显示或打印出来。I表示整数BASIC文件。

5.3.2 文件的存盘、装入和执行

1. 对于不同类型的文件，DOS系统提供不同的命令来操作它们。

对A类文件，DOS提供下述命令：

（1） SAVE 文件名，Ss，Dd

此命令将内存中建好的BASIC文件存到磁盘上。它与文件存磁带命令的区别是：存带命令的文件名要加引号，存盘命令的文件名不加引号。文件名后的Ss和Dd的意义与列目录命令中讲的完全一样。 例如，

］SAVE FILE，S6，D1↙

表示将内存中的文件起名为FILE并存到6号槽口1号磁盘机的磁盘上。注意做SAVE时，磁盘上的写保护标签要拿掉，否则操作会失败。

（2） LOAD 文件名，Ss，Dd

此命令将文件从磁盘装入内存。

例如，

］LOAD FILE↙

表示将文件FILE从磁盘上装入到内存中。

（3） RUN 文件名 Ss，Dd

此命令将文件从磁盘装入内存并执行。如果文件已经用LOAD装入内存或从键盘敲入内存，则可以只敲入RUN再敲回车便可运行程序。

例如：

］RUN FILE↙

即执行文件FILE。

2．对于B类文件，DOS也提供一套命令：

（1） BSAVE 文件名，Ss，Dd，A＄起始地址，L＄长度

此命令将存贮器中的信息按指定的起始地址和长度存入磁盘，起始地址和长度都要求用16进制数表示。例如

]BSAVE ABC, A$1F00, L$2A

表示将从地址$1F00开始长度为$2A，也就是到$1F2A结束的内存单元中的信息（不论是程序还是数据）存到上一次访问过的磁盘上，文件名为ABC。做BSAVE时，软盘上的写保护标签要拿掉。否则，操作会失败。

（2） BLOAD 文件名，Ss，Dd，A$起始地址

此命令将磁盘上文件装入到指定的起始地址上。若不敲入起始地址，则文件装入到BSAVE时给定的地址上。

例如：

]BLOAD ABC↙

将磁盘上文件ABC装入到内存中，装入的起始地址为$1F00，因为BSAVE时是从$1F00开始存盘的。

（3） BRUN 文件名，Ss，Dd

此命令将磁盘上的文件装入内存并执行。

对于整数BASIC程序，要先进入整数BASIC状态，然后才能运行整数BASIC程序。

5.3.3 格式化软盘

在使用一个新软盘时，一般需要先把软盘格式化。格式化的目的是在磁盘上按磁道和扇区来划分地址，并把地址记录在软盘上，便于读写数据时按其地址进行查找。DOS 3.3操作系统没有单独提供格式化命令，但提供了初始化命令。利用初始化命令，可以完成格式化操作。

初始化操作的过程是这样的：首先用DOS系统盘进行启动，然后从盘上装入一个BASIC程序，或从键盘上键入一个新的BASIC程序。移去系统盘，插入空白盘，并关上驱动器门。打入命令：

]INIT 文件名↙

于是磁盘就被格式化。按这种方式格式化的盘上已经装入了DOS系统，并且一开机就自动执行INIT后文件名指定的文件。

5.3.4 拷贝文件和盘

拷贝文件的功能是按文件名将文件从一个磁盘复制到另一个磁盘。拷贝盘的功能是按磁盘上的磁道和扇区将一个磁盘上的全部信息复制到另一个磁盘上。

1. 拷贝文件可以使用LOAD和SAVE命令，即将文件先装入内存，再存到另一个盘上。例如，要把甲盘上的一个文件FILE1拷贝到乙盘上，我们称甲盘为源盘，乙盘为目的盘。甲盘上的文件为源文件，乙盘上的文件为目的文件，操作如下：把源盘插入磁盘机，打入

] LOAD FILE1↙

等又出现提示符] 后，把源盘拿出来，把目的盘插入磁盘机，打入

] SAVE FILE1 ↙

于是文件FILE1存到了目的盘上。

也可以使用DOS实用程序FID来复制文件。FID程序的执行方法如下，打入

] BRUN FID↙

屏幕上将显示选择菜单，也就是这个软件有哪些功能都按顺序列出来的单子，即：

APPLE Ⅱ FILE DEVELOPER

FID VERSION M

COPYRIGHT 1979 APPLE COMPUTER INC.

CHOOSE ONE OF THE FOLLOWING OPTIONS

(1) COPY FILES

（2） CATALOG

（3） SPACE ON DISK

（4） UNLOCK FILES

（5） LOCK FILES

（6） DELETE FILES

（7） RESET SLOT & DRIVE

（8） VERIFY FILES

（9） QUIT

上述菜单中1～9为各种不同的操作，通过键入相应的数字来选择所需要的操作。

敲入1后进入拷贝文件操作，程序提问源文件和目的文件的槽口号、驱动器号以及文件名，在回答文件名时可以使用等于号“＝”作为通配符。即字符“＝”可以代替文件名中的任何字符串，我们把这种字符称为通配符。例如，文件名“A＝”表示所有第一字符是A的文件名。文件名“＝B＝”表示所有文件名中有B字符的文件。文件名“＝”表示所有文件。当使用了通配符后，FID程序会询问。

DO YOU WANT PROMPTING?

即在拷贝时是否再提示一下找到的文件名，由用户决定是否拷贝。

（1） 若回答Y，则在拷贝时每个匹配的文件名就显示在屏幕上，由用户决定是否拷贝，这时若拷贝，再打入一个Y，若不拷，打一个N，若拷了几个文件后不想再继续拷贝，可打入一个Q，结束操作。

（2） 若回答N，则一次将所有匹配的文件都拷贝过去。注意目的盘必须是被格式化过的盘。

2. 对于磁盘拷贝可以使用实用程序COPYA或COPY。

它们分别是用Applesoft BASIC语言和整数BASIC语言编写的。使用时可敲入

] RUN COPYA↙

于是COPYA程序被执行。依次回答拷贝程序提出的问题，如源盘和目的盘的槽号、驱动器号，拷贝工作就开始了。COPYA的一个功能是在拷贝前先自动将目的盘格式化，然后再进行拷贝。

5.3.5 删除文件

有时软盘上的空间没有了，而一些没用的文件占了很多空间，因此想把这些没用的文件删除，腾出地方来存放新的有用的文件。这时就要使用删除文件的命令。

删除文件的命令如下：

] DELETE 文件名，Ss，Dd

其中Ss，Dd的意义与列目录命令CATALOG中的意义一样。例如，

] DELETE FILE2 ↙

将磁盘中文件FILE2删掉。

5.4 使用两个磁盘机进行文件拷贝

使用一个磁盘机进行文件拷贝总要来回换盘，即源盘和目的盘共用一个磁盘机。若有两个磁盘机则可以将源盘放在一个磁盘机中，目的盘放在另一个磁盘机中，不要来回换盘，一次拷完，节省很多时间。

在中华学习机的右上方有一个50线的扩充槽，扩充槽上可以连接一个软盘驱动器卡，此卡可以连接两个磁盘机。注意插卡时元器件一面向着键盘。扩充卡尺寸太大插不进去，

可采用50线转接板把槽口抬高，再插扩充卡。插了扩充卡后，中华学习机就可以连接三个磁盘机。由于在出厂时，扩充槽设置为1号槽口，因此在使用两个磁盘机拷贝时要指明槽口号和驱动器号。例如，我们要将直接与主机连接的磁盘机上的文件拷贝到扩充槽软盘驱动器卡连接的磁盘机上，假定扩充槽上的磁盘机连接在1号驱动器上。于是，命令中应指明源文件在6号槽1号驱动器上，即S6，D1。目的文件在1号槽1号驱动器上，即S1，D1。

5.5 介绍一个游戏程序“击剑”

在这个游戏中，你在屏幕中间，两旁各有一个大汉挥舞着大棒、长枪向你打来。你可以利用手中的剑做武器，步步逼近，向他们发起进攻，击中一人，你就可以得一分。进攻时，你要掌握最佳时机出剑，否则太靠近了容易被大汉打死。同时你还要密切注意随时都可能出现的飞禽走兽。它们将给你造成极大的威胁，使你受伤，减弱进攻能力。甚至使你丧失生命。不过这也不可怕，因为它们都怕你手中的剑！游戏开始时你共有三次生命，在得分达到24分时，还能获得奖励的一次生命。但是如果你被大汉打中了，就丧失一次生命。

在玩游戏时，盘片要一直放在磁盘机中。控制动作的键说明如下：

A—人向左前进	M—剑向下
S—人转身	J—击剑（小动作）
D—人向右前进	L—击剑（大动作）
I—剑向上	K—收剑

第六章　打印机的连接和使用

6.1　扩充槽的使用

主机的右上方有个50线的扩充槽，可插多种外设卡、扩充卡，如打印机卡、驱动器卡、音乐卡等，若扩充卡尺寸太大，插不进去，可采用50线转接板把槽口抬高，再插外设卡或扩充卡。注意插卡时元器件一面向着键盘，要平着向下插。

扩充槽的槽号可以由用户自己设定。在机壳内主机电路板的右侧，有一个短接插头，插头旁边标有相应的槽号，用户可以用短接帽来选择扩充槽的槽号。由于汉字系统使用了 3 号槽，磁盘驱动器接口使用了 6 号槽，因此用户可以选择的槽号为 1 ， 2 ， 4 ， 5 ， 7 中的一个。出厂时一般都设置在第 1 槽，即将两个短接帽插在标有“ 1 ”的位置上。如果用户需要改变槽号，需打开主机壳，把原在 1 槽位置的两个短接帽取下，并插入你所需要的槽号的插头上。

6.2　打印机的连接

先关掉主机和打印机电源，将打印机接口卡插入主机的扩充槽，元器件面向键盘，接口卡通过扁平电缆线与打印机接通，最后接通打印机和主机电源。

6.3　西文方式的打印

在西文方式下，也就是刚一开机时系统所处的状态，使

用打印机输出的操作如下，敲入

] PR#1↲

于是打印机被启动，以后所有显示到屏幕上的信息都同时从打印机上打印出来，直到敲入

] PR#0↲

停止打印机输出。

在程序中也可以用PR#1和PR#0来控制打印机。例如，

```
10  PR#1
20  PRINT"ABCD"
30  PR#0
```

这个程序在运行时先执行PR#1语句，连通打印机，再打印出ABCD四个字符，最后关闭打印机。

6.4 汉字方式的打印

进入汉字方式后，不能使用PR#或IN#这类的语句，因为系统在一个时刻只允许一个槽号与主机相连。进入汉字系统是通过执行PR#3实现的，也就是说主机的中文显示和输入的管理都是通过3号槽上的程序实现的，若此时执行PR#1或IN#1就会使系统脱离中文系统的管理程序。因此，在中文方式下打印输出要使用其他的方法，也就是通过使用BASIC提供的POKE语句直接修改一些专门单元中的值来控制打印机的工作。具体方法如下：

（1） 设置打印方式

格式：POKE 1659, n

其中n=0为不打印，或者说是停止打印。

n＝1～15为打印，并且按系统定义的15种字型进行打印。1659单元是打印方式标志单元，POKE语句的作用是将n送到1659单元中。

（2） 设置字间距

格式：POKE 1787，n

其中n为所打印字符之间的字间距，n取值范围为0～255。即字间距可以在0到255个点之间。刚进入中文系统时，字间距自动设置为1。

（3） 设置行间距

格式：POKE 1915，n

其中1787为行间距标志单元，n为打印行之间的点间距，n取值范围在0到255之间。即行间距可在0到255个点之间，刚进入中文系统时，行间距自动设置为1。

（4） 设置行允许字数

格式：POKE 2043，n

其中n为一行所允许打印的汉字个数，n取值范围可在0～225之间，进入中文系统后，行允许字数自动设置为40。

下面我们编一个打印程序为例。打印程序如下：

```
10  POKE  1659, 1
20  POKE  1787, 3
30  POKE  1915, 8
40  PRINT  "朝辞白帝彩云间"
50  POKE  1659, 2
60  PRINT  "千里江陵一日还"
70  POKE  1659, 3
80  PRINT  "两岸猿声啼不住"
90  POKE  1659, 4
```

```
100  PRINT  "轻舟已过万重山"
110  POKE  1659, 0
```

这个程序的执行过程是这样的：先设置为打印1号字，然后打印第一句诗。再设置为打印2号字，再打印第二句诗。再用3号字打第三句，用4号字打第四句，最后关闭打印机，即可打出四种字体(略)。

6.5 编写图形中有汉字的程序

中华学习机的图象显示区有两页，也就是第一页图象显示区和第二页图象显示区。第一页图象显示区使用内存的$2000到$3FFF这一块存贮区，第二页图象显示区使用$4000到$5FFF这块存贮区。图象显示区的选择是通过BASIC语句HGR和HGR2来实现的，HGR选择第一页，HGR2选择第二页。

由于汉字系统在显示汉字时都是把汉字点阵放在第二页图象显示区，因此，为了在图形上标上汉字说明就必须把图形也画在第二页显示区，即在画图之前先使用HGR2语句。使用HGR2 语句后，不仅可以使屏幕显示使用第二页，还可以清除屏幕，使状态字也清除掉。

在有图形和汉字的时候，可以先显示汉字，再画出图形。这样可以避免在输出汉字时破坏已经显示在屏幕上的图形。如果想先画出图形再输出汉字，则要注意在用PRINT语句输出汉字后用分号结尾。下面是一个图形中有汉字的程序例子：

```
10  HGR2
20  HCOLOR = 3
30  VTAB  5 : HTAB  8 : PRINT"方块"
```

```
40 VTAB 6：HTAB 24：PRINT"三角"
50 HPLOT 47, 50 TO 47, 100 TO 97, 100
   TO 97, 50 TO 47, 50
60 HPLOT 199, 50 TO 159, 110 TO 239,
   110 TO 199, 50
```

或者

```
10 HGR2
20 HCOLOR=3
30 HPLOT 47, 50 TO 47, 100 TO 97, 100
   TO 97, 50 TO 47, 50
40 VTAB 5：HTAB 8：PRINT"方块"；
50 HPLOT 199, 50 TO 159, 110 TO 239,
   110 TO 199, 50
60 VTAB 6：HTAB 24：PRINT"三角"；
70 VTAB 9
```

6.6 屏幕硬拷贝

屏幕硬拷贝是指将屏幕上显示的图形用打印机打出来，也就是将内存图形显示区中的内容全部打印出来。

在西文方式下可以使用下述语句打印屏幕内容：

```
100 POKE 1913, n
110 PRINT CHR $(17)
```

其中n取1为打印第一页图形显示区，n取2为打印第二页图形显示区。

在中文方式下，打印屏幕分两种情况：一种是系统已装入DOS的情况，另一种是没有装入DOS的情况。在有DOS

时，可以使用下述语句：

```
100  PRINT
110  PRINT  CHR$(4); "PR#1": PRINT
120  POKE 1913, 2
130  PRINT CHR$(17)
140  PRINT  CHR$(4); "PR#3": PRINT
```

在没有DOS时可以使用下述语句：

```
100  PR#1: PRINT
110  POKE 1913, 2
120  PRINT CHR$(17)
130  PR#3: PRINT
```

下面是一个打印屏幕的实例，它把前一节介绍的带有汉字的图形打印出来。

```
10  HGR2
20  HCOLOR=3
30  HPLOT 47, 50 TO 47, 100 TO 97, 100 TO
    97, 50 TO 47, 50
40  VTAB 5: HTAB 8: PRINT"方块";
60  HPLOT 199, 50 TO 159, 110 TO 239, 110
    TO 199, 50
70  VTAB 6: HTAB 24: PRINT"三角";
80  VTAB 9
100 PRINT
110 PRINT CHR$(4); "PR#1"; PRINT
115 POKE 1913, 2
120 PRINT CHR$(17)
130 PRINT CHR$(4); "PR#3"; PRINT
```

或者

```
10  HGR2
20  HCOLOR=3
30  HPLOT 47, 5  TO 47, 100 TO 97, 100 TO
    97, 50 TO 47, 50
40  VTAB 5: HTAB 8: PRINT"方块";
60 HPLOT 199, 50 TO 159, 110 TO 239, 110
   TO 199, 50
70 VTAB 6: HTAB 24; PRINT"三角";
80 VTAB 9
90 PR#1: PRINT
115 POKE 1913, 2
120 PRINT CHR$(17)
130 PR#3: PRINT
```

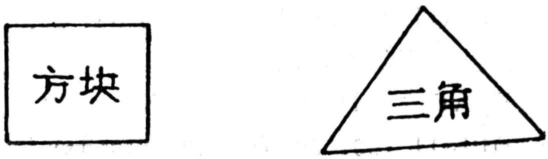


图 6-1

打印出的图形如图6-1。

6.7 介绍一个游戏程序“中国象棋”

人和计算机下棋听起来很吸引人，玩起来更有意思。是人能赢计算机呢还是计算机能胜过人？这需要你自己和计算机下一盘才能知道谁是强者。

这个游戏的玩法说明如下：

程序运行后屏幕出现提示

DO YOU WANT TO PLAY W OR B

(REPLY WOR B)：

这是问你要黑方还是要白方。打W则是要白方，即棋子为黑底白字，并且白方先走。若打B则是要黑方，即棋子为白底黑字，此时，计算机先走，计算机总是为上方。

然后屏幕提示

SELECT LOOK AHEAD：

这是要你选择难度。从1开始共有3级，级别越高难度越大。选择难度之后，屏幕出现棋盘。

走棋方法是打入棋子所在的坐标，然后打“—”，再打入落子点的坐标。坐标顺序是先横向后纵向。横向坐标从左到右为1到9，纵向坐标从下到上为A到J。走棋方法必须符合象棋规则，否则机器将回答：

INVAID MOVE TRY AGAIN。

让你重新走。

使用“ESC”键作为转换开关，进行棋盘显示与走棋步骤表之间的切换。

第七章　LOGO语言

7.1　LOGO语言的介绍

LOGO语言是目前世界上较流行的教学语言。它是美国麻省理工学院的LOGO小组于1968年创立的。这种语言的特点是把画图与程序设计紧密地结合在一起，使程序设计变得充满绘画的乐趣，从而使中小学生们十分感兴趣。不仅如此，LOGO语言还具备结构程序设计语言的品格。作为计算机入门的教学语言，使初学者能较轻快地理解有关计算机的各种基本概念，从而逐步培养应用计算机解决问题的本领。

7.2　LOGO语言的进入和退出

由于CEC-I型中华学习机固化了LOGO语言，因此，一开机打入命令就可以进入LOGO语言，而不用从磁盘机调入到内存。

在没有装入DOS操作系统的情况下，可以直接打入下述命令，

    ] LG↙

系统便进入LOGO语言

若系统已装入DOS操作系统，则要打入下述命令进入LOGO语言：

    ] MAX　FILES1↙

] LG↙

这是因为LOGO语言占内存空间较大，而DOS系统启动后自动设置三个文件缓冲区，这些文件缓冲区将会随着LOGO语言的装入而破坏，因而在LOGO状态下就不能使用DOS的功能了。

MAX FILES1表示将DOS文件缓冲区缩小为1个，再装入LOGO语言后就不会破坏DOS文件缓冲区，因而可以在LOGO状态下使用DOS功能。

当需要退出LOGO语言时，只能按Ctrl—Reset使系统复位。（请参阅第九章）

7.3 LOGO语言的画图功能

LOGO语言设计了一个可以活动的“海龟”，用LOGO的作图命令可以驱使海龟移动，海龟爬行的轨迹就形成了各种图形。屏幕上的海龟实际是一个亮的小三角形“△”光标，只要键入LOGO作图命令，海龟就出现在屏幕的中央，并从这点出发根据命令要求进行活动。

下面是四个基本命令。

命令格式			说明
FORWARD	（或FD）	n	海龟前进n步
BACK	（或BK）	n	海龟后退n步
LEFT	（或LT）	n	海龟左转n度
RIGHT	（或RT）	n	海龟右转n度

上述四个基本命令可以控制海龟的运动方向。每个命令后可跟一个参数，表示前进或后退的长度、左转或右转的角度。由于屏幕上显示的图形都是描点形成直线，再构成图，所以可以把屏幕上点间的距离定作长度单位，简称为

"步"。例如

】FD 100↵

表示前进100步。注意，在FD和100之间必须按一个空格键。空格键后再按参数和回车。于是，我们可以看到海龟从中央位置出发很快向上爬。留下相当于100步长的一条竖线，海龟停留在直线的顶端。如图7-1。



图 7-1



图 7-2

再例如画图7-2的一个方块的程序可以写成：

```
FD    100
RT    90
FD    100
RT    90
FD    100
RT    90
FD    100
RT    90
```

也可以用下述语句：

```
REPEAT 4 [FD 100 RT 90]
```

REPEAT表示重复执行，即重复执行4次前进100步转90度。于是得到图7-2。

除了基本绘图功能外，LOGO语言还提供了各种辅助的绘图语句，如抬笔、落笔、光标回到中心位置、彩色绘图等等。

7.4 LOGO语言的运行方式

LOGO语言有命令方式和程序方式。命令方式就是敲完语句和回车后立即执行语句，语句不保留于内存，如上节的例子就是命令方式。程序方式就是语句保留于内存，不立即执行，需要执行时可以随时调用执行。

LOGO语言的程序结构如下：

TO 过程名

过程体

END

从上面的结构我们可以看出，LOGO语言的程序都是以TO开始，后面是程序名。因为LOGO语言中的程序都可以被别的程序调用，所以程序名也叫过程名。中间是过程体，也就是程序的内容，如FD 100，RT90等语句。最后是END，它表示这个程序结束了。例如：画方块的程序可写成：

```
TO FANGKUAI
REPEAT 4[FD 100 RT 90]
END
```

要执行这个程序时只要敲入过程名再敲回车即可，如敲入

```
] FANGKUAI↵
```

便可画出方块。

在敲完TO和过程名后，LOGO自动进入编辑状态。敲完程序后，敲Ctrl-C便可将程序存入内存并退出编辑状态。如果敲Ctrl—G则退出编辑状态，并且在编辑状态下敲的程序都作废。

下面再看一个画圆的程序。

```
TO YUAN
REPEAT 360 [FD 1 RT 1]
END
```

即重复执行360次，每次走一步转1度，一共走360步转360度，正好转一圈。

7.5 LOGO语言的其他功能

LOGO语言除了提供绘图功能外，还提供了判断功能、计算功能、对字和表的处理功能以及音乐功能。LOGO语言可以支持多层嵌套、过程参数的传递、过程的递归调用。LOGO语言可以使用DOS的磁盘命令，可以将画好的图形存盘或打印。

第八章　监控程序

8.1　监控程序的功能

监控程序是一个管理中华学习机系统的重要程序，它固化在只读存贮器ROM中，一开机，计算机就自动执行监控程序。监控程序的主要功能是管理计算机的键盘输入、屏幕显示、录音机的存取以及提供一套对系统的基本操作，如对存贮器内容的显示、修改、传递、比较，对中央处理器CPU内部寄存器的显示和修改，对内存中机器语言程序的执行、跟踪、反汇编等。它所提供的管理系统的子程序可以为其他程序所调用，减少了编制其它程序的工作量。在中华学习机的监控中还提供了小汇编程序，利用它可以很方便地敲入汇编程序。

8.2　监控命令的使用

8.2.1　进入和退出监控

要使用监控程序的功能首先要从BASIC状态进入监控状态。进入方法很简单，在BASIC下敲入：

］CALL—151↲

便进入监控状态。监控状态的提示符是一个星号“*”。

当需要退出监控回到BASIC状态时，可敲入“Quit”或Ctrl—C键，也可以敲入Ctrl—B键。用“Quit键或Ctrl—C键退出监控时，内存中的程序不会破坏，用Ctrl—B退出监

控时，内存中的程序都被清除。

8.2.2 显示存贮器的内容

在调试程序时，有时要看一看存贮单元中的数据是什么，怎样变化的。这时就可以进入监控状态使用监控程序提供的显示存贮器内容的功能。显示存贮器的命令有四种格式：

格式1：地址↙

第一种格式就是打入你要查看的那个单元的地址，然后回车。“↙”表示回车。这个命令只能显示一个存贮单元的内容。注意地址要用16进制数表示。例如：

```
*1000
1000—20
```

其中20就是＄1000单元中的内容（＄表示16进制数），它是一个16进制数。

格式2：首地址.尾地址↙

第二种格式是将首地址到尾地址的所有存贮器内容显示出来。例如

```
*1000 . 1020
1000—20   00   F8   C4   2C   B0   11   C8
1008—20   0E   F8   90   F6   69   01   48
1010—20   00   F8   68   C5   2D   90   F5
1018—60   A0   2F   D0   02   A0   27   84
1020--2D
```

格式3：尾地址↙

第三种格式是将从上次最后访问过的地址开始到命令中给出的尾地址为止的所有存贮器内容显示出来。例如：

```
*.1040
```

1021—A0　27　A9　00　85　30　20　12
1028—28　F8　88　10　F6　60　48　4A
1030—29　03　09　04　85　27　68　29
1038—18　90　02　69　7F　85　26　0A
1040—0A

因为前面的例子中最后访问的是＄1020单元，所以这个命令显示的是＄1021到＄1040单元中的内容。

格式4：↵

第4种格式只按一下回车键，它是显示从上次最后访问的那个单元开始的8个存贮单元的内容。例如：

1041—18 69 03 29 0F 85 30 0A

8.2.3　修改存贮器内容

当需要直接修改某一个存贮器单元的内容时，可以使用监控程序提供的这个功能，例如要设置一些参数，或修改一个指针等。修改存贮器的命令有两种格式。

格式1：地址：数据　数据……↵

第一种格式是敲入你要修改的存贮单元地址和一个冒号"："，然后就敲你要向这个单元中存放的数据，最后敲回车。如果要连续存放多个数据，可以在每个数据后敲一个空格即可。例如：

＊1000：20　30　40　50↵

这个例子的执行结果是从＄1000单元开始放入＄20，直到＄1003单元放入＄50。

格式2：　数据　数据……↵

第二种格式是从上次用来存放数据的地址之后开始，将数据顺序送入存贮单元。例如：

＊：60　70　80↵

执行后将＄60、＄70、＄80放入了＄1004、＄1005、＄1006。

8.2.4 传送存贮器中的内容

有时需要将存贮器中的一段程序或数据一起搬到另一块存贮区，这时可以使用这个命令。

格式：目的地址＜源地址 1．源地址2 M↲

它的功能是将源地址1到源地址2的内容传送到从目的地址开始的存贮单元中。例如：

＊4000＜1000.2000M↲

将＄1000到＄2000单元中的内容送到＄4000到＄5000中。利用传送命令可以填充存贮区。例如，有时希望将＄1000到＄4000都填上＄FF，可以这样做：先在＄1000中填入＄FF，然后用M命令。

如：

＊1000：FF↲

＊1001＜1000.3FFFM↲

因为传送操作是把＄1000中的内容送到＄1001，＄1001的内容送到＄1002，最后是＄3FFF中的内容送到＄4000，又因为＄1000中已经填入了＄FF，所以最后＄1000到＄4000都填上了＄FF。

8.2.5 比较存贮器中的内容

比较存贮器中的内容是指比较两个数据块的内容。例如，想比较一下两个相似的程序是否完全一样，就可以使用这个命令。

格式：地址1＜地址2．地址3 V↲

命令执行的结果是将地址2到地址3之间的数据与从地址1开始的长度相等的地址范围内的数据进行比较。如果相应

单元内容有不同时，就将相应地址和数据显示出来，若都相同就显示出监控提示符。

例如：

```
*3000<1000.100FV↙
1000—20      (00)
1002—F8      (00)
1003—C4      (00)
1004—2C      (00)
1005—B0      (00)
1006—11      (00)
1007—C8      (00)
1008—20      (00)
1009—0E      (00)
100A—F8      (00)
100B—90      (00)
100C—F6      (00)
100D—69      (00)
100E—01      (00)
100F—48      (00)
```

这条命令是将＄1000到＄100F之间的数据与＄3000到＄300F之间的数据相比较，括号里是＄3000到＄300F的内容。因为＄1001和＄3001的内容一样，所以就没有显示出来。

8.2.6 磁带的读写

在监控状态下，可以将存贮器中的内容存到磁带上，或者将磁带上的内容装入内存。

写带就是把内存的内容写到磁带上，命令格式如下：

首地址·尾地址W↵

即将从首地址到尾地址之间的数据写到磁带上。例如：

*1000.3000W↵

即将＄1000到＄3000的内容写到磁带上。

读带就是从磁带上把数据读到内存中，命令格式如下：

首地址．尾地址R↵

例如：

*2000.4000R↵

将磁带上的内容读到＄2000到＄4000中。注意，读带的首尾地址不一定要和写带时一样，但长度要一样，否则会在读完带做检查时发生错误。

8.2.7 反汇编命令

汇编程序就是用计算机的指令助记符编写的程序，汇编程序经过汇编后产生二进制代码程序，也就是把指令助记符换成二进制的操作码，这样计算机就可以认识和执行了。但是，二进制操作码不容易记住。在调试二进制代码程序时人们希望能看到它们的指令助记符。反汇编命令就提供了这种功能，它把内存中的二进制代码按照地址顺序列出来，而且还在旁边标出相应的指令助记符，这样一看就知道这段程序是在做什么工作了。

反汇编的命令格式如下：

格式1：地址L↵

也就是敲入要查看的地址，然后敲L和回车。反汇编就将从指定地址开始的20条指令翻译成6502指令助记符，并在屏幕上显示出来。

例如：

*1000L↵

```
1000 -  20  00  F8  JSR   $F800
1003 -  C4  2C      CPY   $2C
1005 -  B0  11      BCS   $1018
1007 -  C8          INY
1008 -  20  0E  F8  JSR   $F80E
100B-   90  F6      BCC   $1003
100D-   69  01      ADC   #$01
100F-   48          PHA
1010 -  20  00  F8  JSR   $F800
1013 -  68          PLA
1014 -  C5  2D      CMP   $2D
1016 -  90  F5      BCC   $100D
1018 -  60          RTS
1019 -  A0  27      LDY   #$2F
101B-   D0  02      BNE   $101F
101D-   A0  27      LDY   #$27
101F-   84  2D      STY   $2D
1021 -  A0  27      LDY   #$27
1023 -  A9  00      LDA   #$00
1025 -  85  30      STA   $30
```

其中左边一列是地址；中间是十六进制代码，如第一行的2000F8；右边是指令助记符，如第一行的JSR $F800。

格式2：L↙

第二种格式将从上次最后一个地址的下一个地址开始反汇编20条指令，并显示在屏幕上。

8.2.8 显示和修改CPU寄存器

CPU就是计算机的中央处理器。在调试程序时，有时

需要检查和修改CPU中寄存器的取值。

命令格式：Ctrl—E↙

即先同时按下Ctrl键和E键，抬起后，再敲一下回车键。于是屏幕显示出寄存器的内容，如：

A=30 X=00 Y=00 P=00 S=A0

此时，敲入冒号“：”，然后敲入数据，就可以修改寄存器的值。如

A=30 X=00 Y=00 P=00 S=A0

*：01 02 03 04 05

回车后，A，X，Y，P，S这些寄存器的值就改为1，2，3，4，5。

8.2.9 执行机器指令

假如你用汇编语言编好了程序，经汇编后将目标程序装入内存，你可以用这个命令来执行程序，如果在程序中插入BRK指令，就可以分段执行程序，达到调试程序的目的。

格式1：地址G↙

即从指定地址开始执行机器语言程序，当执行到一条BRK指令时程序停止执行。例如：

*1000G↙

即从$1000单元开始执行程序。

格式2：Ctrl—Y↙

即同时敲Ctrl键和Y键，然后回车，则计算机从$3F8开始执行用户所确定的机器语言程序。通常在$3F8处放置一条JMP指令，转向用户程序的入口。

8.2.10 单步执行

单步执行就是每次执行一条指令，并将指令助记符显示出来，执行后各寄存器的值也显示出来。如果想继续单步执

行下一条机器指令，可以再敲一个S和回车。此过程可以一直进行下去，直到不需要再执行程序时为止。命令格式如下：

地址S↙

例如执行＄1000单元开始的一条指令，并显示寄存器结果和指令助记符：

```
*1000S↙
1000—  A9 55  LDA # $55
A = 55   X = CE   Y = D8   P = 30   S = EC
```

8.2.11 跟踪执行

跟踪执行即是多个单步操作连起来做，命令格式如下：

地址T↙

即从指定地址开始执行机器指令，每条指令执行完后都显示它的助记符和各寄存器的值。直到执行完一条BRK指令为止。如

```
*1000T↙
1000—  A9 55  LDA  # $55
A = 55   X = CE   Y = 55   P = 30   S = F5
1002—  8D 0F 10   STA  $100F
A = 55   X = CF   Y = 55   P = 30   S = F5
1005—  A0 C3  LDY  # $C3
A = 55   X = CE   Y = C3   P = B0   S = F5
1007—  00           BRK
1007—   A = 55   X = CE   Y = C3   P = B0   S = F5
```

8.2.12 16进制加减法运算

监控程序提供了16进制的加减法，但参加运算的数据不能超过两位数，即不能超过＄FF。运算结果也只保留两位

数，即把向第三位的进位丢掉。格式如下：

格式1：数据1＋数据2↙

例如：

＊3A＋4B↙

＝85

格式2：数据1－数据2↙

例如：

＊F4－11↙

＝E3

8.2.13 选择输入、输出设备

在监控下可以选择输入、输出设备。例如希望在监控下打印出屏幕上显示的内容，就可以使用下述命令。

＊1 Ctrl—P↙

这个命令的一般形式是

格式1：槽号 Ctrl—P↙

即将输出控制转给槽号所指定的连接槽上的接口卡。槽号可以取0，1，2，4，5，7。当希望打印机停止打印时，可以用

＊0 Ctrl—P↙

也就是槽号0为返回以监视器为输出设备的状态。

如果希望将输入控制转给槽号所指定的连接槽口上的接口卡，则可以使用下述命令：

格式2：槽号 Ctrl—K↙

当需要返回以键盘为输入设备的状态时，可以打入：

＊0 Ctrl—K↙

8.2.14 多重命令

在输入监控命令时，可以一次输入多个命令，命令之间

用空格分开。最长可以敲入254个字符。例如，把磁带上的程序装入内存然后自动执行，可以打入

*1000.1530R　1000G↵

即将磁带上程序装入＄1000到＄1530，然后从＄1000开始执行程序。

8.3　小汇编的使用

8.3.1　进入和退出小汇编

在监控状态下敲入下列命令行，即可进入小汇编状态：

*D350G↵

这时，屏幕显示小汇编状态的提示符“！”，表示系统已进入小汇编状态。

当需要退出小汇编状态时，只需敲入下列命令：

*＄D360G↵

注意在D360前要先敲一个＄。

8.3.2　敲入指令助记符

在小汇编提示符下，按下列格式工作：

地址：汇编记忆符↵

例如，敲入

！　1000：LDA＄C0 30↵

回车后，屏幕显示为

1000 AD 30 C0 LDA ＄C030

此时，若想继续输入下一条指令助记符，可以先敲一个空格，然后敲助记符，如

1000 AD 30 C0 LDA ＄C030

！ DEY↵

回车后，屏幕显示为

```
1000 AD 30 C0 LDA  $C0 30
1003 88       DEY
```

8.3.3 在小汇编状态下使用监控命令

在小汇编状态下使用监控命令的方法是，在各种监控命令前先敲一个“$”。例如

```
!  $1004<1000.1003M↵
```

于是从$1000开始的两条指令移到从$1004开始的位置。

第九章 中华学习机用法讨论

本章讨论一些用户在实际使用中遇到的问题及CEC-I 1.1版本软件存在的问题和解决方法。

9.1 LOGO语言

第七章已介绍了LOGO语言的进入。下面介绍怎样从LOGO回到BASIC和怎样使LOGO语言显示出彩色。

由于LOGO语言从固件中装入内存后破坏了监控的一些工作单元，并且LOGO语言没有提供退回BASIC的命令。因此即使按下Ctrl-Reset键也不能回到正确的BASIC状态或监控状态。这里介绍一种退回BASIC的方法如下：

第一步敲入·CALL␣64166␣64166↵

第二步敲入Ctrl-Reset

其中“·”是句号,“␣”是空格,“↵”是回车。上面第一步是执行冷启动的初始化操作。第二步是设置正常的显示方式。

在进入LOGO语言时，系统将显示设置为黑白方式。若用户需要控制彩色，可以执行下述操作：

敲入·DEPOSIT␣49328␣16↵；出彩色

或 ·DEPOSIT␣49328␣0↵；关彩色

9.2 汉字方式下的彩色显示

CEC-I中华学习机的彩色显示受到二级控制。第一级控制就是软盘驱动器插座旁边的小开关。此开关在机壳内主板

上。当此开关处于黑白位置时，则显示始终是黑白的。当此开关处于彩色的位置时，显示方式还将取决于第二级控制。这就是程序控制的软开关。在BASIC中，可以用下面的语句控制显示：

```
POKE  49328, 0 ；黑白方式
POKE  49328, 16；彩色方式
```

在汇编程序中，用下面的指令控制显示：

```
LDA   #$00
STA   $C0B0；黑白方式
LDA   #$10
STA   $C0B0；彩色方式
```

由于彩色方式下显示的汉字不如黑白方式下显示的汉字清晰、稳定，因此每次显示汉字时都将彩色关掉，置为黑白方式。这样使汉字显示的效果更好一些。

有些用户希望在彩色图形旁加上汉字说明，这时应如何处理呢？这里提供一个参考方案。当你的程序处于循环读键等待时，你插入一条置彩色的语句。这样，虽然显示汉字时屏幕置为黑白方式，但显示完回到你的程序就立即置成了彩色方式，并且程序在彩色方式下循环等待。那么总的效果就是一幅彩色图形上有汉字说明。只是此时的汉字不如黑白方式下的汉字清晰稳定。若希望汉字本身有颜色，则需要将汉字的点阵作特殊的处理，即放大、加色。实现起来比较复杂。

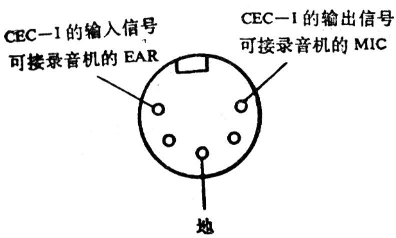
9.3 录音机的接线和使用

这里对录音机的使用作一些补充。

CEC-I中华学习机可接家用录音机，将录音机作为外

存。在使用录音机时应考虑下述情况。因为家用录音机的型号各异，不是专配计算机的磁带机。因此，信号的匹配很重要。通常家用录音机有音量控制旋扭，当音量不合适时，信号不匹配，或者幅度太小，计算机收不到，或者幅度太大，使干扰信号加大，破坏了计算机对正确信号的接收。因此调整音量到合适的位置可能要做几次不同音量的试验，直到计算机接收成功后，将音量旋扭的位置做一个记号，以便下次使用时一次调准。

对于录音机的连接可参考下图：



以上为接录音机的插座。厂家提供的录音机连线可能没有标出哪根线插入录音机的哪个插孔。这时，只要用万用表量一下哪个插头对应五芯插头的哪个插针，并按上图在插头上做一个记号即可。

9.4 中断的使用

CEC-I中华学习机接受非屏蔽中断、请求中断和Reset中断。当中断发生后，CPU的处理过程如下：

对于非屏蔽中断，CPU从地址＄FFFA进入，然后直接转到＄3FB执行。＄3FB是第三页随机存贮器单元，用户可以修改这些单元的内容，使CPU执行用户的中断服务程序。

在开机时，监控的初始化程序已将这些单元填入内容，使控制转到＄FF65，由监控程序处理中断的发生。

对于请求中断，CPU从地址＄FFFE进入，然后转到＄FA40执行一小段程序。这小段程序是判中断源是否是BRK指令产生的。若是就做BRK中断处理，否则进入＄3FE中的地址。在开机初始化时，监控将FF65放入＄3FE，＄3FF。因此用户可以通过修改＄3FE，＄3FF中的内容使CPU执行用户的中断服务程序。

对于复位中断，即Ctrl-Reset中断，CPU从地址＄FFFC进入。然后直接转到＄FA62执行复位操作。

用户在使用中断时应考虑下述问题：

（1）若用户不使用汉字系统，则可直接修改＄3FE，＄3FF的内容，中断发生后进入用户中断服务程序。

（2）若用户使用汉字系统，则启动用户程序后要将扩展RAM区的＄FFFA～＄FFFF填入中断跳转向量地址，即

FFFA：FB03；非屏蔽中断

FFFC：62FA；复位中断

FFFE：40FA；请求中断

注意：用户可以将监控FA40的一小段程序搬到RAM的FA40中，也可将中断处理程序的入口地址直接填入FFFE。

（3）中断发生后程序控制转到中断处理程序。若中断处理程序有切体操作，则应先保存切体状态，即保存C01B到C017最高位的值，然后进行中断服务。返回时，先按保存的状态恢复切体，然后返回。若中断处理程序不改变切体状态，可以不必保存切体状态，直接进行中断服务，然后直接返回。

在中断服务程序中若需要显示汉字，则应先保存汉字字

库的页面状态，页面状态保存在＄EF单元。当中断服务程序显示汉字时，先将＄EF单元的值保存，显示汉字之后，再按＄EF的保存值切体，恢复汉字的页面状态。

9.5 CEC-I 1.1版软件中的若干问题及解决方法

在CEC-I 1.1版软件中有些功能不能实现，有些功能实现得不完美。在CEC-I 1.2版软件中已得到改进。对于1.1版的软件，有一些补救的方法讨论如下。

（1）从光标清到行尾

正常的操作是执行PRINT CHR＄(29)，但不起作用，在BASIC程序中实现这一功能可用下述程序：

```
10  CALL  50091
20  CALL  62348
30  CALL  50105
```

在汇编方式中为：

```
JSR   $C3AB
JSR   $F38C
JSR   $C3B9
RTS
```

（2）HTAB语句

技术参考手册中允许HTAB(x)中的x取值为1～255。实际上西文方式下允许1～255，中文方式下只允许1～34。

第十章　编辑汇编程序的使用

10.1　汇编语言程序概述

计算机的程序设计从使用二进制数字表示的机器语言发展到使用符号表示的汇编语言，之后，又发展到了使用接近人类语言的高级语言。程序设计语言的迅速发展和更新换代大大推动了计算机技术的普及和应用。目前，在中小学学生中首先学习了高级语言BASIC。这对于迅速了解和掌握计算机的程序设计无疑是很有效果的。在学会了BASIC语言之后为什么还要学习汇编语言呢？要回答这个问题就要先了解汇编语言的特点和优点。

汇编语言是一种直接与计算机指令系统密切相连的计算机语言。它直接用计算机指令的汇编符号来编写程序。此外，汇编语言还提供一些伪指令。这些伪指令在写程序时可以象指令一样地被使用。这些伪指令中，有的是对汇编过程（将汇编语言源程序转换成机器码表示的目的程序，这个过程称为汇编）本身进行控制或提供一定参数；有的是为程序的执行提供数据区或数据；还有的是对目的程序的属性以及与其他程序的关系等做出规定。使用汇编符号表示的指令和伪指令，按照一定的格式和语法关系写的程序称为汇编语言源程序。用系统提供的软件汇编器，或称汇编程序，便可将这种源程序汇编成目的程序。

汇编语言比机器语言前进了一大步。它的优点在于：

（1）机器语言是用二进制数字表示指令的，汇编语言是用汇编符号表示指令的，因此比较直观，容易记忆。

（2）在用机器语言写程序时必须计算出程序分支转移的地址，而汇编语言用标号代替地址，写程序时可不考虑转移地址的具体数值。

（3）程序修改较方便。不会因为修改一条而牵动整体。

（4）允许操作数为简单的表达式，或为标号，这给编写程序提供了方便。

（5）允许对源程序进行注解，便于阅读、理解。

在有了伪指令以后，汇编语言又增加了其他优点。例如对保留内存数据，保留某些特殊字符等都可以通过使用伪指令实现。

汇编语言和高级语言相比，虽然汇编语言仍属低级语言，使用不方便，比较繁锁，并且与机器的硬件密切相关，但它却有下面的特点：

（1）节省内存空间和CPU资源。

（2）执行速度较快。

（3）能准确地掌握程序执行时间。

由于这些特点，在许多情况下，例如实时控制系统和数据采集系统等都需要使用汇编语言来编写程序。另外，一些应用软件和系统软件的开发也是使用汇编语言比较合适。

计算机系统除了提供汇编器外，为了便于用户编写汇编语言源程序，也都提供编辑程序。利用编辑程序，用户可以方便地编写和修改汇编语言源程序。还可以方便地存储、装入、打印汇编语言源程序。

美国苹果公司为苹果Ⅱ机提供了多种汇编程序。6502编辑/汇编程序是最常用的汇编工具软件。6502编辑/汇编程序

主要可分成三个部分：编辑程序、汇编器和编辑汇编命令解释程序。由于中华学习机兼容苹果Ⅱ机，因此苹果Ⅱ机上的软件可以直接在中华学习机上运行。

10.2 汇编过程简介

上面说过，6502编辑/汇编程序是由三部分组成，即编辑部分、汇编部分和命令解释执行部分。编辑部分的主要功能是帮助使用者建立一个源程序。汇编部分的主要功能是将源程序汇编成为目的程序。命令解释部分的主要功能是统一分析执行编辑程序和汇编程序的命令，使编辑和汇编溶为一体，便于用户的使用。编辑功能的使用将在后面介绍，这里先介绍汇编的过程。

汇编是采取二次扫描的办法。即，汇编器对源程序从头到尾“阅读”两次。第一次阅读后，主要是产生一个符号表，第二次才产生出整个目的程序。这种“阅读”是以磁盘中边读入内存边进行的。在第二次扫描中则同时把产生的目的程序按256个字节为一段，逐段地写到磁盘上形成目的程序文件。在扫描过程中会把错误信息打出来（指出那些行是错的）。如果没有错误，机器最后给出“NO ERROR”的信息。

为什么两次扫描可以从源程序产生出目的程序来呢？这就要了解这两次扫描都做了哪些事情。

汇编器中主要存放有两个表格。一个是机器操作码表MOT(Machine Operation Table)；一个是伪指令操作表POT(pseudo Operation Table)。另外还有一个内存地址计数器LC (location Counter)。在第一次扫描过程中，LC从0开始不停地追踪源程序行，对每一行均根据机器

操作码表和伪指令操作表求出其在目的程序中应占的单元数，根据此数修改地址计数器（不停地加算得的数）。当遇到有标号时，便把LC当时的值赋给它。因此在第一次扫描后，各标号的值便可确定下来了，这样，便可产生一个符号表，对所有的符号和标号均赋好了值。而第二次扫描时，再次逐条查找MOT表和pOT表。当遇到操作数中有标号时，便可从符号表中查出它的值代入。这样，便可最后形成目的程序。

下面举例说明两次扫描怎样形成目的程序。首先给出一个源程序。它的功能是将16个数据构成的数据块从内存单元DATA～DATA十15移到0页的＄F0～＄FF。源程序如下：

```
1                ORG     $8000
2   BUFFER       EQU     $F0
3                LDX     #0
4   START        LDA     DATA, X
5                STA     BUFFER, X
6                INX
7                CPX     #16
8                BNE     START
9                RTS
10  DATA         DS      16
```

我们先分析一下几个伪指令的作用。第一行中的“ORG ＄8000”即表示下面的程序是产生绝对目的地址的，其起始的内存单元地址为＄8000。第二行为一等值语句，即表示下面的源程序中符号BUFFER就等于＄F0。第十行的伪指令表示在这里开始保留16个单元（但暂时不存放任何信息）。

现在看一下第一次扫描时会产生什么结果。开始LC＝0，但扫到第1行时，它便从伪指令操作表中查到“ORG”的含意，就会自动将LC置为$8000。扫第2行时，又是一个伪指令，它不影响LC，而只在符号表中建立起第一个符号值“BUFFER $F0”。扫第3行时，从机器指令操作表中得知“LDX #0”为两字节指令。LC便自动加2。变成$8002。扫到第4行时，由于有标号“START”，而LC＝$8002，便将$8002这个值赋给标号“START”。这是符号表中的第2行。同时看到DATA是个前面没有赋过值的符号，汇编器便把它当做一个16位的地址值来计算。因而“LDA DATA，X”这条指令肯定是3字节指令。这样LC加3变成$8005。当扫到第5行时，由于“BUFFER”这个符号已定义为$F0，可以知道此指令应为两字节，所以LC变为$8007。以下类推。在最后一行得到“DATA”为$800D。这样，在第一次扫描中便建立起了符号表。并且源程序中的所有标号和符号都确定了具体值。符号表为：

```
BUFFER   $F0
START    $8002
DATA     $800D
```

在第二次扫描时，便可根据机器指令操作表和伪指令操作表以及刚产生的符号表把所有源程序中的指令汇编成机器语言的数码了。下面是机器对上述子程序汇编后打印出的最后结果。前面是目的程序，后面是符号表。源程序文件名为“MOVE”；目的程序名为“MOVE·OBJ0”

```
SOURCE  FILE: MOVE
......       NEXT  OBJECT  FILE NAME  IS
              MOVE.OBJ0
```

```
8000:                1            ORG      $8000
00E0                 2  BUFFER EQU         $F0
8000 : A2  00        3            LDX      #0
8002 : BD  D0  80    4  START     LDA      DATA, X
8005 : 95  F0        5            STA      BUFFER,X
8007 : E8            6            INX
8008 : E0  10        7            CPX      #16
800A : 0D  F6        8            BNE      START
800C : 60            9            RTS
800D :              10  DATA      DS       16
* * *  SUCCESSFUL ASSEMBLY : NO ERRORS
F0   BUFFER  800D  DATA  8002  START
```

从上述过程中还可看到，在汇编源程序中，对于0页的标号或单字节符号（例如上述的“BUFFER”）都应在引用它们之前定义。如果放在后面，例如上例中的“BUFFER EQU $F0”放在第5行之后，则第1次扫描时将按3字节来处理“LDA BUFFER, X”这条指令。这种引用为“向前引用”（确定其值之前已引用）。对于两字节标号，向前引用是允许的。

10.3 6502编辑/汇编程序的进入和退出

一般情况下，编辑/汇编程序的制造厂商提供一张软盘，上面有编辑/汇编程序和DOS系统。

如果直接从这张软盘上启动，即将此盘插入驱动器，打开电源开关（若已打开电源开关则可敲入“PR # 6”），则在操作系统引导进入内存后，便会自动将编辑汇编程序的命令解释程序和编辑程序装入内存。汇编程序仍留在软盘上，待使用汇编命令时再装入内存。

在装入命令解释程序和编辑程序时，屏幕会显示标题，如：

APPLE II EDITOR-ASSEMBLER

CURRENT ASSEMBLER ID STAMP IS:

20 JAN 89 #000001

最后一行称为ID标记，等待你对它进行修改。此时可输入你要修改的年、月、日及后面的数（此数随着每次进行汇编时而自动增加）。若不修改，则直接打回车，系统便会自动进入命令方式，其提示符号为冒号“：”。在此方式下，你可以建立汇编语言源程序，也可以调用汇编器。

如果操作系统已经调入内存，可用下述命令进入编辑/汇编程序：

RUN EDASM

如果不想修改上述标记，也可用下述命令直接调入：

BRUN EDASM•OBJ

从编辑/汇编程序的命令方式转到BASIC或监控状态，可以分别使用命令END和MON，也可以从这两个状态再回到编辑/汇编的命令方式。这些转换命令可表示于图10-1中。

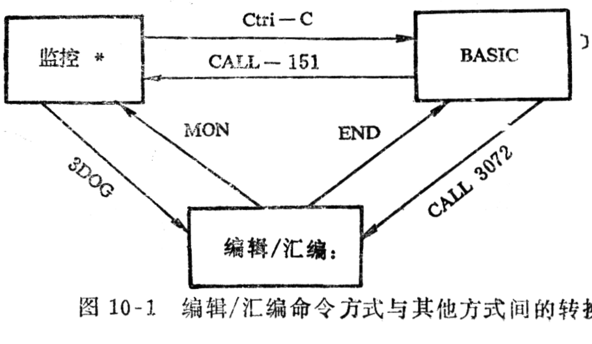


图 10-1 编辑/汇编命令方式与其他方式间的转换

注意，在监控和BASIC状态下输入其他信息，会破坏保存在内存中的汇编源程序。CALL3072是将盘上的编辑程序重新装入内存。

10.4 编辑/汇编提供的命令

在命令方式下，编辑/汇编程序共提供31个键盘命令。这些命令可分为三类：一类是编辑命令，这部分命令主要用来建立汇编源程序。第二类是操作命令，这部分命令是在编辑过程中提供一些辅助手段。例如，控制源程序所在的内存区域，了解源程序的存放情况，了解磁盘中的文件目录，文件的保存和装入，等等。第三类只有一个命令，就是汇编命令，对指定的源程序进行汇编。

在进入命令方式后，用户可以打入问号和回车使所有的命令都显示在屏幕上。显示的内容如表10-1所示。

表10-1 编辑/汇编命令表

```
ASM  SOURCE FILE, 〈OBJECT FILE〉, S〈#〉, D〈#〉
APPEND  〈LINE#〉SOURCE FILE
ADD 〈LINE#〉
CATALOG
COPY  LINE# -〈LINE#〉TO LINE#
CHANGE  〈BEGIN#〈-END#〉〉.OLDSTR.NEWSTR.
DRIVE#
DELETE  〈BEGIN#〈-END#〉〉
END
EDIT  〈BEGIN#〈-END#〉〉, STRING
FILE
FIND  〈BEGIN#〈-END#〉〉, STRING
HIMEM= #
INSERT LINE#
LENGTH
```

续表

LO MEM = #
LOAD SOURCE FILE
LIST 〈BEGIN#〈-END#〉〉
MON
NEW
PR##, 〈DEVCTL〉
PRINT 〈BEGIN#〈-END〈#〉〉
REPLAGE 〈BEGIN#〈-END#〉〉
SLOT#
SAVE 〈BEGIN#〈-END#〉〉 〈SOURCE FILE
TRUNC OFF
TRUNC ON
TLOAD 〈BEGIN#〈-END#〉〉
TSAVE 〈BEGIN#〈-END#〉〉
TABS 〈BEING#〈-END#〉〉.STRING.
WHERE LINE#

表10-1中每行的第一组字符是命令本身，带有下划线的字符在屏幕上是反相显示的，使用时可以省略。“SOURCE FILE”是指源程序文件名，“OBJECT FILE”是目的程序文件名。除了“PR##，〈DEVCTL〉”这个命令中的第一个#号外，其他#号均表示此处应输入一个十进制数字，“#”本身并不要打入。“LINE#”为编辑方式下源程序中的行号，“BEGIN#”和“END#”分别表示起始行行号和结束行行号。“.STRING.”表示带有分隔符的字符串（此处分隔符用的是“.”，若标号中使用了“.”，分隔符可改用“/”）。括号〈 〉中的内容为命令中可有可无的部分。例如：

：ASM␣TEST，TST1，D2↵ （↵表示回车）

这个命令表示对文件名为TEST的汇编语言源程序进

行汇编。产生的目的程序文件命名为TST1，存放在2号磁盘驱动器的磁盘上。这里省掉了“S#”，这说明源文件所存的磁盘插在与上述2号磁盘驱动器属于同一插槽的磁盘驱动器上。如果希望目的文件和源文件在同一个磁盘上，且不指明目的文件名，可以敲入：

：ASM　TEST↙

则目的文件名为TEST.OBJ0，且和源文件在同一个磁盘上。这个命令的详细用法将在后面说明。

再例如：

：C5-20.LDA.LDX.↙

这个命令表示，它要求把现在内存中的源程序的第5至20行中所有的字符串“LDA”均变为“LDX”。如果在上述命令中没有5到20这个行号范围，则要求对整个源程序中的字符串“LDA”均改为“LDX”。

10.5　源文件的建立

10.5.1　源程序的格式

利用编辑程序建立一个源程序，目的是通过汇编器产生一个目的文件，这个目的文件是6502 CPU 可执行的机器语言程序。源文件被规定为由若干个逻辑记录构成。每个逻辑记录均以回车字符为结束。在内存中，逻辑记录由最高位为1的ASCII码表示的字符串组成，我们称一个逻辑记录为一行，或一个语句。逻辑记录的一行在屏幕上可能不止一行，由回车符来确定。但一行不能超过256个字符。

为了能汇编出目的程序，每个逻辑记录必须满足下面的条件：

（1）满足汇编程序对它的格式要求。

（2）所使用的指令和伪指令必须是符合规定的。

对于6502的指令，这里不作详细介绍。可参见6502的指令集，伪指令将在后面详述。

汇编程序规定，源程序的每个行可分成四个字段。即：标号字段、操作码字段、操作数字段和注释字段。

对每一字段又分别有对它们的具体格式和内容方面的规定。

（1） 标号字段

任何非纯注释语句均可有标号字段。标号必须是用A到Z的英文大写字母之一作开头。其中只准含有字母、数字和句号，不准含有空格。总计字符最多可有16个。

标号必须从一行开头处开始，它必须是唯一的。即在一个源程序中一个标号只能定义一次。即使一个源程序包含多个文件也须在各文件间遵守这一规定。

有的特殊语句必须有标号，例如“EQU”伪指令为操作码字段的行就是如此。

（2）操作码字段

操作码字段必须是指令的汇编符号（即指令助记符）或是伪指令的汇编符号。操作码字段与标号字段间必须只留一个空格。若前面没有标号则与行起始处只留一个空格。

（3）操作数字段

操作数字段的有无取决于操作码字段是什么。有的操作码要求后跟操作数，有的则不要。

操作数字段是一个表达式，它可由常数、标号以及运算符+、－、*（乘）、/（除）构成。所谓常数包括下述四种：

① 十进制常数：0～65535

② 十六进制常数：$0～$FFFF（以$开头）

③ 八进制常数：@0～@177777（以@开头）

④ 串型常数：串型常数是一串字符，其两端要用单引号“'”括起来。汇编时，将取字符的ASCII码来作为它的数值。在不作特殊规定时，此ASCII码的最高位取为1。例如“A”汇编为$C1（A的ASCII码为$41）。由于6502的立即数是8位二进制数，不允许超过$FF。所以串型常数作操作数字段时，仅第一个字符是有效的。例如“LDA 'A'”这条指令经汇编后成为“A5C1”，即相当于“LDA #$C1”。而“LDA'ABC'”这条指令同样也汇编出同一结果。

表达式中的运算符无优先级的规定。自左至右顺序运算。表达式计算结果不能超过16位二进制数所表达的范围，否则，汇编器将指出溢出错误。当表示一个8位立即数时，结果按256取模。

（4） 注释字段

注释字段是可选择的字段。汇编时对它是忽略的。只是在列源程序时将它们与前面的字段一起列出。它可以是任何一组ASCII字符。在它前面的操作数或操作码字段有明确的结束点时，它前面可以没有空格（见后面的例句）。

除了四个字段要按上述规定写以外，还要注意以下几点：

① 各字段间必须仅以一个空格分开。

② 允许一行中只有注释部分，此时可以用“＊”或“；”开始。这样的语句在汇编时将被忽略，不产生目的程序。

③ 空白语句是非法的。即不允许一行中没有任何字

符。

④ 有些字段可以省略，但字段间的空格一般不能省略。操作数字段在有明显的结束点时，它和注释字段间的空格可以省略。

⑤ 一行最多只能包括256个字符。当输入字符超过248个时，机器发出“嘟嘟”声响以提醒。

根据这些规定，下面的语句是正确的：

〈1〉 START␣LDA␣#0␣; A GETA ZERO

〈2〉 ␣INX␣; X ADD ONE

〈3〉 LOO␣AND␣#$FF

〈4〉 *THIS PROGRAM RESET THE SYSTEM

〈5〉 ␣ASC␣'THE CONTROL SYSTEM

〈6〉 BUFFER␣DS␣ 128

〈7〉 ␣LDA␣#@377

〈8〉 ␣ROR␣$A89F; THE RESULT ROTATES RIGHT

其中“␣”表示空格。第4例是一纯注释语句。5、6两例是伪指令语句。第8句“;”前空格的省略是允许的。

下面的例子是错误的：

〈1〉 LOOP （无操作码字段）

〈2〉 ␣ORA （无操作数）

〈3〉 EOR␣#$FF （标号字段位置无空格）

10.5.2 伪指令

共有23个伪指令。这里选择常用的介绍如下。

（1）汇编控制伪指令

① ORG （定起始值）

格式：ORG 表达式

此语句的功能是建立目的程序的起始地址。此语句允许有标号。汇编程序中若无此语句，则不产生目的程序文件，而只产生一个列表。因此ORG语句一般在源程序的开始。例如：

ORG　$1000

此语句表示以下的源程序建立的目的程序在内存的起始地址为$1000单元。

一般情况下，一个源程序中只有一个ORG语句。如果出现第二个，则从第二个ORG语句开始产生另一个目的程序文件。例如：

```
ORG  $1000
     ⋮        第一个目的程序
ORG  $2000
     ⋮        第二个目的程序
```

上述形式为ORG的绝对形式。还有一种相对形式。它不产生另一个目的程序文件，只修改目的程序的地址值。其格式为：

ORG＊＋|表达式|

或　ORG＊－|表达式|

例如："ORG＊＋30"只将当前地址值加30。

又如在下例中：

```
              ⋮
$801  LOOP  LDA #0
$803        STA DATA, X
$806        INX
$807        BNE LOOP
$809        RTS
```

```
        ORG   * + $100
$90A  DATA DS   128
          ⋮
```

假定标号LOOP为＄801，则DATA这一语句原应为地址等于＄80A（假定DATA为非0页地址）。但遇到上面的ORG伪指令后，它变为＄90A。标号DATA赋值为＄90A。

② EQU （等值伪指令）

格式：标号 EQU 表达式

此语句用来给标号指定一个值。这个语句中必须要有标号。并且不允许此标号再作其他语句的标号使用。

例如：

```
DISPLAY  EQU  $FDF0
              ⋮
         JSR  DISpLAY
              ⋮
```

这个例子经过汇编则DISPLAY被赋值为＄FDF0这个地址值，而“JSR DISPLAY”便被汇编为“JSR ＄FDF0”。当程序中有的标号是外部模块的子程序，而本模块希望自己单独汇编调试时，经常使用“EQU”来定义外部模块子程序标号的地址值。当地址值需要改动时，只要修改EQU中的地址值即可．本模块中所有调用此标号的指令都不用改动。

③ CHN （连接）

格式：CHN 源文件名 〈，S#〈，D#〉〉

这条伪指令表示当前的源程序与CHN中指出的源文件相连接。此语句一般是放在源程序的最后一行。例如：

; ------------------------

```
;FILENAME: FILE1
;-------------------------
        ORG    $1000
START   LDA    #$0B
         ⋮
         ⋮
END     RTS
        CHN   FILE2
```

这个源程序在汇编时将会自动与FILE2源程序一起汇编为一个目的程序。FILE2这个程序接在现在这个程序FILE1的后面。

这个语句的提供，可以使程序设计人员把程序分成一个一个模块，汇编时连在一起汇编。修改时，分开修改。节省程序存盘取盘的时间。

④ MSB （置最高位）

格式： MSB ON

或者 MSB OFF

此命令是控制汇编产生的ASCII码的最高位为1或0。当有“MSB ON”时，后面的字符串所产生的ASCII码的最高位均为1。而“MSB OFF”指令的后面产生的ASCII码的最高位都为0。但这条命令对下面要介绍的DCI伪指令不发生影响。

当无MSB命令时，即相当于有“MSB ON”的情况。这是因为，在学习机中，当ASCII码的最高位为1时，正好显示时对应的是正常显示方式；而最高位为0时，则对应于反相显示方式。

（2） 数据定义伪指令

① ASC （字符串定义 1 ）

格式：ASC 有分隔符的字符串

在此伪指令中，有分隔符的字符串指的是一个字符串首尾有分隔符。所谓分隔符指除去逗号（，）、连接号（-）、数字、空格和回车符以外的任一符号，但在该字符串本身中不能再含有此符号。例如：

'SUN AND MOON'

]ABCDEFG]

HSOFTWAREH

其中“'”、“]”、“H”都是分隔符，当后面没有注释字段时，仅用前面的分隔符便可，后面的可省略。

这个命令用来在内存中存放该字符串的ASCII码。ASCII均为8位，其高位取决于此语句前面最接近的MSB语句。若无MSB语句时最高位为1。字符串码存放的起始单元即为当前汇编器地址指针的值。此语句允许有标号。例如：

TITLE ASC 'EDUCATION COMpUTER'

② DCI （字符串定义 2 ）

格式：同ASC

此伪指令定义的字符串在汇编时不受MSB指令的影响。所定义的字符串中，前面的字符串高位都为0，最后一个字符高位为1。在读取这种字符串时，可以通过判最高位是否为1来确定字符串是否读完。

③ DFB （定义字节）

格式：DFB 表达式〈，表达式…〉

此指令的功能类似于前面两条指令。它定义的不是ASCII字符而是表达式所表示的数值。数值的范围为0～255。超过范围按256取模。

例如：

```
DATA  DFB  0, 1, 2
      DFB  $0F, $BE
```

④ DW （定义字）

格式：DW 表达式

此指令定义一个16位二进制数，即两个字节长度的数。其低8位在第一字节，高8位在第二字节。例如：

```
WROD  DW  $FD1B
```

此时，在WORD单元存入$1B，在WORD+1单元存入$FD。WORD是标号，汇编时赋于地址指针的值。

当定义的数值是一个地址且地址值在汇编时才能确定时，可以放入地址的标号。例如：

```
              ⋮
SUBROUTIN   LDA  0
              ⋮
WORD        DW   SUBROUTIN
              ⋮
              ⋮
```

汇编后，标号STBROUTIN的地址值自动放入WORD和WORD+1单元。

⑤ DDB （定义双字节）

格式：与DW一样

此指令也定义一个16位二进制数，与DW的区别是高8位在前一单元，低8位在后一单元。例如：

```
WORD  DDB  $IBFD
```

此语句与WORD DW $FDIB完全一样。

⑥ DS （保留存储单元）

格式：DS　表达式

这个伪指令的效果是从当前地址单元保存内存某些字节，保存的个数即为表达式的值，也允许有标号。这个伪指令可用来保留内存工作单元或数据区。

除上述数据定义伪指令外，汇编程序对地址标号和立即数的表示有如下规定：

标号地址的低八位用“#>”表示，高八位用“#<”表示。例如：

```
LDA  #>TABLE
STA  $F0
LDA  #<TABLE
STA  $F1
```

其中“#>TABLE”指标号TABLE的低八位。而“#<TABLE”指标号TABLE的高八位。

立即数用“#”表示。如

```
LDA  #9
LDX  #$FB
```

其中第一条指令是取立即数9，“#”表示立即数，9为十进制数。第二条指令是取立即数$FB，“#”表示立即数，$表示十六进制数。

（3）条件汇编

条件汇编共有三个伪指令，它们必须配合使用。

① DO　（条件汇编命令）

格式：DO　表达式

这个命令表示根据表达式的值是否非0来确定紧跟其后的条件汇编语言块是否进行汇编。

② FIN　（条件汇编结束命令）

格式：FIN

此指令表示条件汇编语言块的结束。

③ ELSE（汇编条件取反命令）

此指令只能放在条件汇编语言块之中。例如：

```
DO  表达式
    ⋮ } 块 A
ELSE
    ⋮ } 块 B
FIN
```

此时把条件汇编语句块分成两个部分，我们称它们为块A和块B。当DO语句表达式为0时，块A不进行汇编，但块B要进行汇编；反之，若表达式非0，则块A汇编而块B不汇编。

（4）列表命令

列表命令是用来对汇编时的输出列表的格式进行控制，对目的程序无影响。列表命令共有6种。

① PAGE （换页）

格式：PAGE

当遇到程序中有此命令时，汇编程序送给输出设备（如打印机）一个换页命令，同时送给显示器回车换行命令，使屏幕上显示一个空行。它不产生任何目的程序。

② LST （列表）

格式：LST ON

或 LST OFF

当汇编程序遇到“LST OFF”时，它在第二次扫描中

对后面的行不输出列表。即不打印也不显示。而遇到“LST ON”时则相反。源程序中无任何列表命令时，列表将自动地进行。

利用此命令可以对列表的部分加以控制。比如只想打印或显示目的程序中的一部分，就可以用此命令实现。

③ SKP （跳行）

格式：SKP 表达式

汇编列表输出时，遇到此命令时列表设备将列出一些空行。空行数等于表达式的值（以256取模）。

④ SBTL （副标题）

格式：SBTL 有分隔符的字符串

此命令使汇编输出列表时每一页给出一个标题，标题的内容便是命令中提供的字符串。同时每页第一行还给出ID标记，然后打出页数。在打符号表时也给出一标题：“SYMBOL TABLE SORTED BY SYMBOL”和“SYMBOL TABLE SORTED BY ADDRESS”。

⑤ REP （重复字符串）

格式：REP 表达式

汇编遇到这一伪指令时，输出一个重复的字符行。所重复的字符在不规定时为“*”。也可以用CHR命令来规定。字符重复个数等于表达式的值（以256取模）。这个伪指令提供了一种分割子程序或程序模块的表示方法。

⑥ CHR （重复字符）

格式：CHB ？字符

此指令要与上面的REP连用。可规定REP引起的重复字符串中用什么字符。例如：

⋮

```
15      CHR   ? #
16      REP   12
17      CHR   ? $
         ⋮
         ⋮
30      REP   15
         ⋮
```

假设18到29之间再无其他CHR语句，则输出列表时第16和30行分别为12个“#”号和15个“$”号：

```
         ⋮
16  # # # # # # # # # # # #
         ⋮
30  $ $ $ $ $ $ $ $ $ $ $ $ $ $ $
         ⋮
```

CHR语句中必须有问号，否则将按重复字符为空格来处理，即印出一个空行。

10.6 汇编命令和汇编过程的控制

10.6.1 汇编命令

当源程序建立好之后，就可以对源程序进行汇编。汇编命令的格式是：

ASM 源文件名 〈，目的文件名〉〈，S#〉，〈D#〉

其中目的文件名可以省略。当省略时，产生的目的文件名为“源文件名·OBJ0”。若源文件中有多个ORG指令，则汇编结果产生出多个目的程序。目的程序文件名将依次为

源文件名·OBJ0

源文件名·OBJ1

源文件名·OBJ2

⋮

S＃和D＃表示目的文件存放的槽口号和驱动器号。S＃和D＃也可以省略，当省略时，目的文件存放在源文件所在的槽口号和驱动器号上。

源文件必须存放在调用ASM命令时编辑程序所在的磁盘驱动器中。在执行ASM命令时，系统首先调入汇编器。然后给出信息：

PRESS ANY KEY TO CONTINUE

此时，必须将有源文件的盘插入到上述编辑程序所在的磁盘驱动器中。如果该驱动器正好是存放编辑/汇编程序者，而盘上又无该源文件，则要取下该盘，换上有该源文件的盘，然后再按一下任何键便开始汇编。当汇编结束时，又自动出现上述信息，此时可再将有编辑/汇编程序的盘片换回去，按任一键，回到命令状态。

例如，系统在6号槽上有两个磁盘驱动器D1和D2。编辑/汇编程序和源文件是在D1上，希望目的文件放在D2上，可敲入：

ASM ZHUO, RAN, S6, D2

于是汇编源文件ZHUO，将目的文件RAN存于2号驱动器上。又如：

ASM JIXIA

此命令将源文件JIXIA汇编成目的文件，目的文件名为JIXIA·OBJ0（假设只产生一个目的程序文件），仍存放在源文件所在的磁盘上。

10.6.2 汇编过程的控制

当屏幕显示

PRESS ANY KEY TO CONTINUE

之后又敲入任何一键时，汇编过程开始。这时，如果不加干预，将自动汇编，直到结束。但在汇编过程中，有三种命令可对汇编过程进行控制。

（1）停止汇编，回到编辑状态（Ctrl-C）

在汇编过程中，出现汇编错误，或需要马上停止汇编时，可敲入Ctrl-C。此时，系统将关闭所有打开的文件，释放正在使用的磁盘操作系统缓冲区。

（2）暂停汇编（空格键）

在汇编的第二次扫描过程中，源程序清单和目的程序代码会显示到屏幕上或打印到纸上，按下空格键可使汇编暂停，便于观看屏幕上的信息。然后按任意其他键时又继续汇编。如果按下的又是空格键，则显示一行后又暂停。

（3）关闭显示和打开显示

在汇编过程中，为加快汇编速度可关闭显示。打入Ctrl-N就可以关显示及打印，Ctrl-O又开显示及打印。这两个键盘命令的作用和源程序中的“LIST OFF”和“LIST ON”的作用一样。一般在较长的程序进行汇编时才使用这两个命令。比如在一个长程序进行汇编时，只想看其中一小部分，便可用这两个命令来控制。

10.7 源程序的编辑

源程序的编辑包括两方面的内容。一个是对源文件内容的修改，这需要一套编辑命令。另外，在编源文件时需要对源文件进行一些操作。如将文件存盘或装入内存，将两个文件合并成一个文件，查找源文件中程序行对应的地址，等等。

10.7.1 编辑命令

共有十个编辑命令可以在命令方式下直接使用。

（1） 插入命令（INSERT）

格式：INSERT行号

其中有下划线的字符可省略。行号在1～65519的十进制整数范围内。此命令表示从源程序的某一行开始插入。例如，在建立一个新文种时可敲入：

I 1↙

于是源程序就从第一行开始建立。编辑程序接到此命令后，将显示行号1，然后可开始输入源程序。每输入一个回车就自动增加一行，列出新的行号。如果内存中已有部分源程序，插入以后，原来的行顺序向后移。例如：

I 150↙

这是在内存中第150行处开始插入程序。插入后，第150行及以后的行都自动后移 。

从插入方式下退出的方法是敲Ctrl-D或Ctrl-Q，再按一下回车。例如在上例中，若想插入2行（150～151），则当显示到152时插入结束要退出；敲Ctrl-Q和回车，退到命令方式。内存中的源程序则插入了2行。原来的第150行自动变为第152行。

（2） 增加命令（ADD）

格式：ADD〈行号〉

此命令用来对内存源文件增加一些行，功能和插入命令类似，只是插入点有一点差别。如果此命令不带行号，则从最后加起，如果有行号则从该行号的下一行加入。因此，如果行号是10，则“A 10”命令是从第11行开始加入，而“I 10”是从第10行开始插入。如果内存中没有源文件，则“A↙”

从第一行加起。退出的方法和插入命令一样。

（3）替换命令（CHANGE）

格式：CHANGE〈行号1〈-行号 2〉〉 字符串1 字符串 2

此命令的功能是用一个新字符串即字符串 2 替换老字符串即字符串 1 。字符串 1 和字符串 2 的开始处均应有分隔符。字符串 2 的结尾处分隔符可有可无。分隔符的规定与数据定义伪指令ASC中介绍的相同。替换的范围由命令中给出的行号决定。行号 1 是起始行，行号 2 是结束行。如果只有行号1，就只对行号 1 进行操作。如果有行号 1 和“-”，就对从行号 1 到源文件结尾行进行操作。如果没有行号，就对源文件的所有行进行操作。

例如：

C200·LDA·LDX·↵

此命令的意思是将200行中的LDA换成LDX。

又如：

C250-·LABLE1·LABLE2↵

此命令将把250行以及其后的所有行中的LABLE1 改为LABLE2。

再如： C/KSW·CHN/KSW·ASC↵

此命令把所有行中的标号KSW·CHN换成KSW·ASC。

当执行替换命令时，系统先提示下述字符：

ALL OR SOME?（A/S）

如果希望一次都替换好，打入A（或除S和上档键以外的任何字符），系统便自动完成命令所要求的替换，替换的行被显示出来，然后回到命令状态。如果希望有选择地替换，则敲入S（或任何一个上档键）。此时，系统按顺序每次找到一个应该替换的行便停下来，将该行显示在屏幕上，等待输入。

如果要替换，可按空格键(或其他非控制键、非回车键)；如果不想替换该行，可按回车键(或ESC键或除Ctrl-C以外的任何控制键)。

如果在上述过程中希望取消替换命令，可按Ctrl-C

在字符串1中可以使用“Ctrl-A”字符作为通配符。即表示任何一个字符。例如：

C·KEYIN(Ctrl-A)·KEY↙

此命令是将所有行中字符串“KEYINX”换成“KEY”。这里X是任意字符。

（4）复制命令(COPY)

格式：COPY　行号1-〈行号2〉TO 行号3

此命令可以起到复制或“搬家”的作用。其中行号及连接号的规定同前。这个命令将行号1到行号2的源程序复制到从行号3开始的行中去。被复制的部分仍存在源程序中。如果希望把一段源程序搬到另一个地方，可以利用此命令先复制过去，然后再把被复制的部分删去即可。

当行号3小于行号1或2时，这部分程序要后移。例如：

CO　5-10　TO　3

命令执行后，新的源程序中3～8行的内容为原来5～10行的内容，但原5～10行的内容后移到11～16行。

（5）删除命令(DELETE)

格式：DELETE　行号1〈-行号2〉，…

此命令的功能是删除源程序行。允许一次删除多个行的范围，但要十分小心，不要删错，因为在删除过程中，每删一段，后面的行号就顺序向前提，原来的行就变了。例如：

D　2-4，8-10↙

执行结果将删去2～4和11～13行。因为删去2～4行后，原来的11～13行变成了8～10行。

删除的范围不允许出现只有前面的行号和连接号，而无后面的行号。如：

D 10-20，25，29-

其中“10-20”是删除10到20行。“25”就是25行，但“29-”是不允许的。

（6）列表命令(LIST)

格式：LIST 〈行号1〈-行号2〉〉，…

此命令将源程序显示在屏幕上。若没有行号，则列出源程序的所有行。若有行号范围，则只列出范围内的行。对行号的规定与删除命令相同，但是允许使用“行号1-”的形式。

在列表过程中可以用Ctrl-C来中止，回到命令状态。也可以用空格键暂停，然后用除Ctrl-C以外的键恢复。如果连续按空格键，则为逐行列表的方式了。如果希望连续使用相同的列表命令，可以用Ctrl-R来代替上次用过的列表命令。

（7）不带行号的列表(PRINT)

格式：PRINT 〈行号1〈-行号2〉〉，…

此命令与LIST相同，仅显示时不显示行号。

（8）置换命令(REPLACE)

格式：REPLACE〈行号1〈-行号2〉〉

这个命令实际上等效于删除命令和插入命令的相加。例如：

R 200-205

这个命令的执行结果等于下面两个命令执行的结果：

D 200-205↙

I 200↙

（9）查找命令(FIND)

格式：FIND 〈行号1〈-行号2〉〉，字符串

此命令中对于行号的规定与LIST命令的规定相同。字符串前面必需有分隔符。

此命令的功能是在给出的行号范围内查找命令中给出的字符串。被查到的字符串显示在屏幕上。字符串中允许使用Ctrl-A来代表任一字符。

（10）编辑命令(EDIT)

格式：EDIT〈行号1〈-行号2〉〉，字符串

此命令中行号的规定同前。

此命令的功能是在行号指出的范围中对含有该字符串的行进行编辑。执行此命令后系统进入逐行编辑的方式，例如：

E 10↙

则编辑第10行，编辑后立即返回命令状态。而

E 10-↙

则从第10行开始编辑，逐行列出源程序行。然后可逐行修改。下面的命令：

E 10-30，50-60·LOOP↙

则使系统把10～30和50～60这些行中含有字符串“LOOP”的行逐行地列出以进行编辑。字符串中允许使用“Ctrl-A”来表示任一字符。

当进入逐行编辑方式时，便可进行插入、修改和删除字符等操作。先用右箭头键移动光标，使光标停在要进行修改的字符位置上，然后进行插入、删除等操作。操作的方法如

下：

① 插入字符：先打Ctrl-I，然后打插入的字符，这些字符便会插在光标指向字符的前面。要停止插入，可用箭头键或除了Ctrl-V以外的任何控制键。

② 删除字符：敲入Ctrl-D，光标处的字符便被删掉了。

③ 替换字符：直接敲入字符键便可。

④ 插入控制字符：若要插入控制字符，则打入Ctrl-I之后，在每个控制字符之前还需再打入一个Ctrl-V。

⑤ 删除控制字符：删除控制字符与删除普通字符是一样的，即用Ctrl-D。但控制字符平常是不显示出来的。有时汇编指出某行有错，但用LIST命令看不出有错，而在编辑方式下，当光标正好移到控制字符所在的位置时，控制字符才显示出来，它是以闪烁方式出现的。当光标移走时，控制字符也就消失了。

⑥ 替换控制字符：打入Ctrl-V，然后打控制字符。

⑦ 查找字符：当打入Ctrl-F再打入一要查找的字符后，系统便从当前光标处向右查找，找到第一个该字符时，将光标停在那里。若找不到，则仍停在原处。例如，源程序行为：

```
100  CSW  STA  ACC; SAVE A
```

光标在C处，打入Ctrl-F E后，光标移到E字符上：

```
100  CSW  STA  ACC; SAVE A
```

但如果打入的是Ctrl-F B，则光标不动，因为此行中没有B。

⑧ 恢复原行：在对一行进行编辑时，希望取消前面已经进行了的修改而恢复修改前的行，可打入Ctrl-R，这时便

可重新对该行进行编辑。

⑨ 退出编辑：退出编辑有三种方法。第一种是打回车键。此时不管光标停在何处都结束该行的编辑，进入下一行的编辑。如果该行是最后一行，则回到命令状态。

第二种是打入Ctrl-T。此时光标前的字符有效，光标处及其后的字符均删除了。

第三种是打入Ctrl-X。前两种方式都进入下一行的编辑，只有当该行是行范围中的最后一行时才回到命令状态。而Ctrl-X是立即回到命令状态，不再继续编辑。

10.7.2 操作命令

（1）装入文件（LOAD）

格式：LOAD 源文件名

此命令将磁盘中的源文件装入内存。

（2）保存文件（SAVE）

格式：SAVE〈行号1〈-行号2〉〉〈源文件名〉

此命令将行号1到行号2范围内的行保存到当前指向的磁盘上。并命名为指出的文件名。如果无行号，则保存所有源程序内容。

如果执行此命令前已执行过“LOAD”或“SAVE”命令，则文件名可省略。系统将自动命名它为上次访问过的文件名。即先删除盘上该文件名的内容，将内存中的新内容写入。

（3）增补文件（APPEND）

格式：APPEND 〈行号〉 源文件名

此命令是将磁盘上源文件读入内存，接到指出的行号处。内存中原该行以后的行不被后移，而是被取代。如果没有行号，则接在内存中源文件最后一行之后。此命令可将两

个源文件连成一个源文件。

（4）设置插槽号（SLOT）

格式：SLOT 插槽号

此命令设置当前系统指向的磁盘插槽号。如果无此命令，系统指向上次访问的磁盘插槽号。

（5）设置驱动器号（DRIVE）

格式：DRIVE 驱动器号

此命令设置当前系统指向的驱动器号。

（6）列目录

格式：CATALOG

此命令列出磁盘驱动器上文件目录。

（7）列内存中文件名（FILE）

格式：FILE

这个命令显示出最新访问过的文件名，以及它所在的插槽号、驱动器号、文件长度及内存余下的字节数等。

（8）设置地址上界（HIMEM）

格式：HIMEM＝地址值

此命令可设置源程序文件可使用的最大地址。地址值要用十进制数。如果不使用此命令，系统自动把上界定在38400（相当于$95FF），这是最大允许值。一般不必使用此命令。

（9）设置地址下界（LOMEM）

格式：LOMEM＝地址值

此命令与上一命令相对应。通常系统自动置为8192（相当于$2000）

（10）显示源文件长度（LENGTH）

格式：LENGTH

此命令显示目前源程序文件所使用的内存长度及余下的存储空间。

（11）截短源文件（TRUNCATE）

格式：TRUNC ON

或 TRUNC OFF

输入第一个命令后，由LIST或PRINT命令引起的列表将把源程序截短，即不显示注释字段。打入第二个命令后又恢复到不截短的状态。

（12）查找地址（WHERE）

格式：WHERE 行号

执行此命令将显示出内存中存放该行源程序的第一个单元的地址值。

（13）打印源程序（PR #）

格式：PR # 插槽号，〈设备控制字符串〉

此命令是使系统和其他外设相连。通常是与打印机相连，打印出汇编的内容和符号表。设备控制字符串对打印机进行控制，也即使打印机初始化。设备控制字符串分为两部分。前面一部分说明打印的逻辑页长度和物理页长度；后面一部分说明一行打印多少列。逻辑页长度指实际上一页打印多少行，它是用两位十进制数前面放一个L来表示的。物理页长度指一页共有多少行，即这一页顶部到下一页顶部之间的行数，它是两位十进制数前面放一个P来表示。如果不说明则逻辑长度自动设置为60，物理页不说明则在使用SBTL时，每页打完后自动走纸。

例如，通常插槽号为1，打入下述命令：

PR # 1，L54P65 Ctrl-I 80N↙（Ctrl-I不显示）

当用ASM命令进行汇编时，打印机将逐段地随汇编过程打

出汇编内容。打印方式为每页66行而只打印54行。每行最多打80个字符。

在调程序时不必按页打印，而源文件通常也不会在一行超过80个字符，因此只要打入PR＃1↙即可。

（14）格式化命令（TABS）

格式：TABS 数1〈，数2，…〉〈字符〉

此命令是用来对列表命令（LIST和PRINT）产生的列表格式进行控制的。如果不使用此命令，系统置于6502汇编语言的标准格式下。即

行号	␣	标号字段	␣	操作码字段	␣	操作数字段	␣	注释字段
0—4	5	6——13	14	15——18	19	20——28	29	30——

其中“␣”为空格。

若不想使用此格式，可以使用TABS命令。命令格式说明中的“数1，数2，…”指各字段间格式化标记的位置。而〈字符〉指出格式化标记使用的字符。若不说明〈字符〉则以空格字符作为格式化标记。由于汇编只认空格为格式化标记，所以〈字符〉通常都不使用。此命令一般仅为几个递增的数字。这些数字要在0～39之间才有效。最多可有16个。例如：

T 14，19，29↙

此即为上述6502汇编的标准格式。即列表时第一个空格固定在14列，第二个在19列，第三个在29列（行号与标号间的空格是固有的，不能规定）。若使用命令：

T 14•H↙（字符前要有分隔符）

则列表时在14列以前遇到H字符时，便把H列在第14列，然后向右排下去。

（15）刷新命令（NEW）

格式：NEW

此命令并不将内存中源文件清除掉。而仅仅将内存源文件的结束点置为LOMEM命令指定的值（若没使用 LOMEM则为＄2000）。此时再用列表命令时，将什么也不列出来。但是，如果想恢复内存源文件，可以将上述结束点置回原来的值。这个值存在0页＄0E和＄0F单元中。＄0E为低位，＄0F为高位。可以将系统转到监控状态，恢复此二单元原先的值，之后再回到编辑/汇编的命令方式下时，便可列出源文件。

（16）磁带读写（TLOAD和TSAVE）

格式：TLOAD　行号

和　TSAVE　〈行号1〈-行号2〉〉

此二命令用来读写磁带。第一条命令将使磁带中的源文件从指出的行号开始插入。第二条命令将内存中源程序内容按指出的行范围存入磁带中。并允许有多个行号范围。

由于没有磁盘子系统就无法运行编辑/汇编程序，因此这两条命令很少使用。

10.7.3　命令方式下的其他作用

（1）操作系统命令的使用

在命令方式下可以使用某些操作系统命令。方法是先打入一个句号再打入操作系统命令。例如：

·RENAME　文件名1，文件名2 ↵

但要注意，有些命令会破坏编辑/汇编程序，出现不可预测的结果，因此建议只使用下述操作系统命令：RENAME，DELETE，LOCK，UNLOCK，PR＃1等。

（2）多个命令的输入

多个命令可以在一行中输入，而命令之间用命令分隔符

分隔。分隔符为冒号“：”。如：

D 5 - 8 ：C 5 - 8 ·A·B：L↙

这条命令先删除第5～8行，再把5～8行中的A变为B，再把所有源程序列出来。

要用这种方式调用ASM时，ASM必须为最后一个命令。

若不想用冒号作为命令分隔符，可以改变。在命令方式下按一下“ESC”键，然后按冒号，再按想用的分隔符，此时若按下空格键，系统会发出嘟嘟声而等待输入另一字符。如果按下回车键则会显示出当前的命令分隔符。命令分隔符除了不能用空格和回车符外，也不能用句号。

附　　录

附录A　助记符指令表，状态寄存器

6502微处理器指令系统——按字母顺序

ADC	带进位加法
AND	逻辑与
ASL	右移
BCC	进位为“0”分枝，或小于分枝
BCS	进位为“1”分枝，或大于等于分枝
BEQ	结果为“0”分枝
BIT	位测试
BMI	结果为负分枝
BNE	结果不为“0”分枝
BPL	结果为正分枝
BRK	强制暂停
BVC	溢出为“0”分枝
BVS	溢出为“1”分枝
CLC	清进位标志
CLD	清十进制方式
CLI	清中断禁止位
CLV	清溢出标志
CMP	累加器与存储器比较
CPX	变址寄存器X与存储器比较
CPY	变址寄存器Y与存储器比较
DEC	存储器减1

DEX	变址寄存器X减1
DEY	变址寄存器Y减1
EOR	逻辑异或
INC	存储器加1
INX	变址寄存器X加1
INY	变址寄存器Y加1
JMP	转移
JSR	转子
LDA	送累加器
LDX	送变址寄存器X
LDY	送变址寄存器Y
ISR	右移
NOP	空操作
ORA	逻辑或
PHA	累加器压入堆栈
PHP	处理器状态压入堆栈
PLA	堆栈弹回累加器
PLP	堆栈弹回处理器状态
ROL	循环左移
ROR	循环右移
RTI	中断返回
RTS	子程序返回
SBC	带进位的减法
SEC	置进位标志
SED	置十进制方式
SEI	置中断禁止位
STA	存累加器
STX	存变址寄存器X
STY	存变址寄存器Y

TAX	累加器送变址寄存器X
TAY	累加器送变址寄存器Y
TSX	栈指针送变址寄存器X
TXA	变址寄存器X送累加器
TXS	变址寄存器X送栈指针
TYA	变址寄存器Y送累加器

状态寄存器的内容：

7	6	5	4	3	2	1	0
S	V		B	D	I	Z	C

C—进位状态（来自第7位的进位）

Z—零状态（1表示结果为零，0表示结果非零）

I—中断屏蔽状态（1表示中断被屏蔽）

D—十进位方式状态（1表示十进位方式）

B—中止状态（1表示已执行一条Break指令）

V—溢出状态

S—符号状态（等于第7位的值）

注意：第5位没有使用。

附录B 带有操作码、执行周期和存储要求的指令表

常用标志含义如下：

A	累加器
X，Y	变址寄存器
M	存储器
P	处理器状态寄存器
S	堆栈指针
✓	改变
–	不改变

+	加
∧	逻辑与
−	减
∀	异或
↑	弹出堆栈
↓	压入堆栈
→	传送到
←	传送到
∨	逻辑或
PC	程序计数器
PCH	程序计数器高位
PCL	程序计数器低位
OPER	操作数
#	立即寻址方式

ADC 带进位加

N Z C I D V

操作：A＋M＋C→A，C　　　✓ ✓ - - - ✓

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	ADC# oper	66	2	2
零页	ADC oper	65	2	3
零页X变址型	ADC oper，X	75	2	4
绝对型	ADC oper，	6D	3	4
绝对X变址型	ADC oper，X	7D	3	4*
绝对Y变址型	ADC oper，Y	79	3	4*
变址间接型	ADC （oper，X）	61	2	6
间接变址型	ADC （oper，Y）	71	2	5*

• 如果有跨页则加一个周期

AND 逻辑与

累加器逻辑与

操作：A∧M→A

N Z C I D V
✓ ✓ - - - -

寻址方式	汇 编 语 言 格 式	操作码	字节数	周期数
立即型	AND# oper	29	2	2
零 页	AND oper	25	2	3
零页X变址型	AND oper, X	35	2	4
绝对型	AND oper	2D	3	4
绝对X变址型	AND oper, X	3D	3	4*
绝对Y变址型	AND oper, Y	39	3	4*
变址间接型	AND (oper), X	21	2	6
间接变址型	AND (oper), Y	31	2	5*

* 如果有跨页则加1个周期

ASL 左移

操作：C ← | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | ← 0

N Z C I D V
✓ ✓ ✓ - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
累加器型	ASL A	0A	1	2
零页	ASL oper	06	2	5
零页X变址型	ASL oper, X	16	2	6
绝对	ASL oper	0E	3	6
绝对X变址型	ASL oper, X	1E	3	7

BCC 进位为“0”分枝

N Z C I D V

操作：C＝0 分枝　　　　　　　　　　　　　　　　　　　　　　— — — — — —

寻址方式	汇编语言格式	操作码	字节数	周期数
相对型	BCC oper	90	2	2*

* 如果分枝发生在同页上则加1。
* 如果分枝发生在不同页上则加2。

BCS 进位为“1”分枝

N Z C I D V

操作：C＝1 分枝　　　　　　　　　　　　　　　　　　　　　　— — — — — —

寻址方式	汇编语言格式	操作码	字节数	周期数
相对型	BCS oper	B0	2	2*

* 如果分枝发生在同一页上则加1。
* 如果分枝发生在下一页上则加2。

BEQ 结果为“0”分枝

N Z C I D V

操作：Z＝1 分枝　　　　　　　　　　　　　　　　　　　　　　— — — — — —

寻址方式	汇编语言格式	操作码	字节数	周期数
相对型	BEQ oper	F0	2	2*

* 如果分枝发生在同一页上则加1。

如果分枝发生在下一页上则加2。

BIT 位测试

操作：$A \wedge M$，$M_7 \to N$，$M_6 \to V$

位6和位7被传送到状态寄存器

如果$A \wedge M$的结果是零则$Z=1$，否则$Z=0$。

N	Z	C	I	D	V
M_7	✓	-	-	-	M_6

寻址方式	汇编语言格式	操作码	字节数	周期数
零　页	BIT oper	24	2	3
绝对型	BIT oper	2C	3	4

BMI 结果为负分枝

操作：N=1 分枝

N	Z	C	I	D	V
-	-	-	-	-	-

寻址方式	汇编语言格式	操作码	字节数	周期数
相对型	BMI oper	30	2	2*

* 分枝发生在同一页上则加1。
* 分枝发生在不同页上则加2。

BNE 结果不为“0”分枝

操作：Z=0 分枝

N	Z	C	I	D	V
-	-	-	-	-	-

寻址方式	汇编语言格式	操作码	字节数	周期数
相对型	BNE oper	D0	2	2*

* 如果分枝发生在同一页上则加1。
* 如果分枝发生在不同页上则加2。

BPL结果为正分枝

NZCIDV

操作：N=0 分枝

\- - - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
相对型	BPL oper	10	1	2*

* 如果分枝发生在同页上则加1。
* 如果分枝发生在不同页上则加2。

BRK 强制暂停

NZCIDV

操作：强制中断 PC+2↓P↓

\- - - 1 - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	BRK	00	1	7

* BRK命令不能用置I位屏蔽

BVC 溢出为“0”分枝

NZCIDV

操作：V=0 分枝

\- - - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
相对型	BVC oper	50	2	2*

* 如果分枝发生在相同的页则加1。
* 如果分枝发生在不同的页则加2。

BVS 溢出为"1"分枝

操作：V=1 分枝

N Z C I D V
\- - - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
相对型	BVS oper	70	2	2*

* 如果分枝发生在同一页上则加1。

* 如果分枝发生在不同页上则加2。

CLC 清进位标志

操作：0→C

N Z C I D V
\- - 0 - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	CLC	18	1	2

CLD 清十进制方式

操作：0→D

N Z C I D V
\- - - - 0 -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	CLD	D8	1	2

CLI 清中断禁止位

操作：0→I

N	Z	C	I	D	V
－	－	－	0	－	－

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	CLI	58	1	2

CLV 清溢出标志

操作：0→V

N	Z	C	I	D	V
－	－	－	－	－	0

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	CLV	B8	1	2

CMP 累加器与存储器比较

操作：A—M

N	Z	C	I	D	V
✓	✓	✓	－	－	－

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	CMP #oper	C9	2	2
零页	CMP oper	C5	2	3
零页X变址型	CMP oper，X	D5	2	4
绝对型	CMP oper	CD	3	4
绝对X变址型	CMP oper，X	DD	3	4*
绝对Y变址型	CMP oper，Y	D9	3	4*
变址间接型	CMP (oper，X)	C1	2	6
间接变址型	CMP (oper)，Y	D1	2	5*

* 如果跨页则加1。

CPX 变址寄存器X与存储器比较

	N	Z	C	I	D	V
操作：X—M	✓	✓	✓	-	-	-

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	CPX #oper	EO	2	2
零页	CPX oper	E4	2	3
绝对值	CPX oper	EC	3	4

CPY 变址寄存器Y与存储器比较

	N	Z	C	I	D	V
操作：Y—M	✓	✓	✓	-	-	-

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	CPY #oper	C0	2	2
零页	CPY oper	C4	2	3
绝对型	CPY oper	CC	3	4

DEC 存储器减1

操作：M—1→M

N Z C I D V
✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
零　页	DEC　oper	C6	2	5
零页X变址型	DEC　oper，X	D6	2	6
绝　对	DEC　oper	CE	3	6
绝对X变址型	DEC　oper，X	DE	3	7

DEX 变址寄存器减1

操作：X—1→X

N Z C I D V
✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	DEX	CA	1	2

DEY 变址Y寄存器减1

操作：Y—1→Y

N Z C I D V
✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	DEY	88	1	1

EOR 逻辑异或

操作：A∀M→A

N Z C I D V
✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	EOR #oper	49	2	2
零页	EOR oper	45	2	3
零页X变址型	EOR oper, X	55	2	4
绝对	EOR oper	4D	3	4
绝对X变址型	EOR oper, X	5D	3	4*
绝对Y变址型	EOR oper, Y	59	3	4*
变址间接型	EOR (oper, X)	41	2	6
间接变址型	EOR (oper), Y	51	2	5*

* 如果跨页则加1

INC 存储器加1

操作：M+1→M

N Z C I D V
✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	节字数	周期数
零页	INC oper	E6	2	5
零页X变址型	INC oper, X	F6	2	6
绝对	INC oper, X	EE	3	6
绝对X变址型	INC oper, X	FE	3	7

INX 变址寄存器X加1

N Z C I D V

操作：X+1→X　　　　✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	节字数	周期数
隐含型	INX	E8	1	2

INY 变址寄存器Y加1

N Z C I D V

操作：Y+1→Y　　　　✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	INY	C8	1	2

JMP 转移

N Z C I D V

操作：(PC+1)→PCL　　　　- - - - - -

(PC+2)→PCH

寻址方式	汇编语言格式	操作码	字节数	周期数
绝对型	JMP oper	4C	3	3
间接型	JMP (oPer)	6C	3	5

JSR 转子

操作：PC＋2↓，(PC＋1)→PCL
(PC＋2)→PCH

N Z C I D V
\- - - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
绝对型	JSR oper	20	3	6

LDA 送累加器

操作：M→A

N Z C I D V
✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	LDA #oper	A9	2	2
零页	LDA oper	A5	2	2
零页X变址型	LDA oper, X	B5	2	4
绝对型	LDA oper	AD	3	4
绝对X变址型	LDA oper, X	BD	3	4*
绝对Y变址型	LDA poer, Y	B9	3	4*
变址间接型	LDA (oper,X)	A1	2	6
间接变址型	LDA (oper),Y	B1	2	5*

* 如果跨页则加1

LDX 送变址寄存器X

N Z C I D V

操作：M→X ✓✓- - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	LDX #oper	A2	2	2
零页	DDX oper	A6	2	3
零页Y变址型	LDX oper, Y	B6	2	4
绝对型	LDX oper	AE	3	4
绝对Y变址型	LDX oper, Y	BE	3	4*

* 当跨页时则加1。

LDY 送变址寄存器

N Z C I D V

操作：M—Y ✓✓- - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	LDY #oper	A0	2	2
零页	LDY oper	A4	2	3
零页X变址型	LDY oper, X	B4	2	4
绝对型	LDY oper	AC	3	4
绝对X变址型	LDY oper, X	BC	3	4*

* 当跨页时加1。

LSR　右移

M或A

操作：0 → | 7 | 6 | 5 | 4 | 2 | 1 | 0 | → C

N	Z	C	I	D	V
0	√	√	-	-	-

寻址方式	汇编语言格式	操作码	字节数	周期数
累加类型	LSR　A	4A	1	2
零　　页	LSR　oper	46	2	5
零页X变址型	LSR　oper, X	56	2	6
绝 对 型	LSR　oper	4E	3	6
绝对X变址型	LSR　oper, X	5E	3	7

NOP　空操作

操作：空操作(2周期)

N	Z	C	I	D	V
-	-	-	-	-	-

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	NOP	EA	1	2

ORA　逻辑或

操作：AVM—A

N	Z	C	I	D	V
√	√	-	-	-	-

寻址方式	汇编语言格式	操作码	字节数	周期数
立 即 型	ORA　#oper	09	2	2
零　　页	ORA　oper	05	2	3
零页X变址型	ORA　oper, X	15	2	4

（续）

寻址方式	汇编语言格式	操作码	字节数	周期数
绝对型	ORA oper	0D	3	4
绝对X变址型	ORA oper, X	1D	3	4*
绝对Y变址型	ORA oper, Y	19	3	4*
变址间接型	ORA (oper, X)	01	2	6
间接变址型	ORA (oper), Y	11	2	5*

* 有跨页时则加1。

PHA 累加器压入堆栈

操作：A↓

N Z C I D V

\- - - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	PHA	48	1	3

PHP 处理器状态压入堆栈

操作：P↓

N Z C I D V

\- - - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	PHP	08	1	3

PLA 堆栈弹回累加器

操作：A↑

N Z C I D V
√ √ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	PLA	68	1	4

PLP 堆栈弹回处理器状态

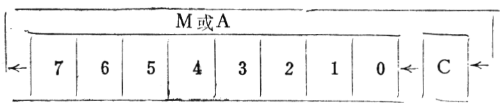
操作：P↑

N Z C I D V
来自堆栈

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	PLP	28	1	4

ROL 循环左移

操作：

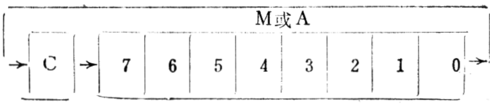


N Z C I D V
√ √ √ - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
累加器型	ROI A	2A	1	2
零 页	ROI oper	26	2	5
零页X变址型	ROI oper, X	36	2	6
绝 对 型	ROI oper	2E	3	6
绝对X变址型	ROI oper, X	3E	3	7

ROR 循环右移

操作：



N Z C I D V
√√√- - -

寻址方式	汇编语言格式	操作码	字节数	周期数
累加器型	ROR A	6A	1	2
零　页	ROR oper	66	2	5
零页X变址型	ROR oper, X	76	2	6
绝对型	ROR oper	6E	3	6
绝对X变址型	ROR oper, X	7E	3	7

RTI 中断返回

N Z C I D V

操作：P↑PC↑　　来自堆栈

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	RTI	40	1	6

RTS 子程序返回

N Z C I D V

操作：PC↑，PC+1→PC　　- - - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	RTS	60	1	6

SBC 带借位的减法

操作：$A-M-\overline{C}\rightarrow A$

注：$\overline{C}$借位

N	Z	C	I	D	V
√	√	√	-	-	√

寻址方式	汇编语言格式	操作码	字节数	周期数
立即型	SBC #oper	E9	2	2
零页	SBC oper	E5	2	3
零页X变址型	SBC oper, X	F5	2	4
绝对型	SBC oper	ED	3	4
绝对X变址型	SBC oper, X	FD	3	4*
绝对Y变址型	SBC oper, Y	F9	3	4*
变址间接型	SBC (oper, X)	E1	2	6
间接变址型	SBC (oper), Y	F1	2	5

* 当跨页时则加1。

SEC 置进位标志

操作：$1\rightarrow C$

N	Z	C	I	D	V
-	-	1	-	-	-

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	SEC	38	1	2

SED 置十进制方式

操作：$1\rightarrow D$

N	Z	C	I	D	V
-	-	-	-	1	-

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	SED	F8	1	2

SEI 置中断禁止位

操作：1→I　　　　N Z C I D V
　　　　　　　　　— — — 1 — —

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	SEI	78	1	2

STA 存累加器

操作：A→M　　　　N Z C I D V
　　　　　　　　　— — — — — —

寻址方式	汇编语言格式	操作码	字节数	周期数
零　页	STA oper	85	2	3
零页X变址型	STA oper, X	95	2	4
绝　对	STA oper	8D	3	4
绝对X变址型	STA oper, X	9D	3	5
绝对Y变址型	STA oper, Y	99	3	5
变址间接型	STA (oper, X)	81	2	6
间接变址型	STA (oper), Y	91	2	6

STX 存变址寄存器X

NZCIDV

操作：X→M　　　　　　　　　　　　------

寻址方式	汇编语言格式	操作码	字节数	周期数
零　页	STX oper	86	2	3
零页Y变址型	STX oper, Y	96	2	4
绝对型	STX oper	8E	3	4

STY 存变址寄存器Y

NZCIDV

操作：Y→M　　　　　　　　　　　　------

寻址方式	汇编语言格式	操作码	字节数	周期数
零　页	STY oper	84	2	3
零页X变址型	STY oper, X	94	2	4
绝对型	STY oper	8C	3	4

TAX 累加器送变址寄存器X

NZCIDV

操作：A→X　　　　　　　　　　　　√√----

寻址方式	汇编语言格式	操作码	字节数	周期数
隐含型	TAX	AA	1	2

TAY　累加器送变址寄存器Y

操作：A→Y　　　　N Z C I D V
　　　　　　　　　√ √ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	TAY	A8	1	2

TYA　变址寄存器Y送累加器

操作：Y→A　　　　N Z C I D V
　　　　　　　　　√ √ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	TYA	98	1	2

TSX　栈指针送变址寄存器X

操作：S→X　　　　N Z C I D V
　　　　　　　　　√ √ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	TSX	BA	1	2

TXA 变址寄存器X送累加器

N Z C I D V

操作：X→A ✓ ✓ - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	TXA	8A	1	2

TXS 变址寄存器X送栈指针

N Z C I D V

操作：X→S - - - - - -

寻址方式	汇编语言格式	操作码	字节数	周期数
隐 含 型	TXS	9A	1	2

附录C　字母、字符显示码表(基本字符组)

	反向显示		闪烁显示		正向显示		正向显示	
	大写字母	特殊字符	大写字母	特殊字符	大写字母	特殊字符	大写字母	小写字母
	$00-$1F	$20-$3F	$40-$5F	$60-$7F	$80-$9F	$A0-$BF	$C0-$DF	$E0-$FF
00	@		@		@		@	、
01	A	!	A	!	A	!	A	a
02	B	"	B	"	B	"	B	b
03	C	#	C	#	C	#	C	c
04	D	$	D	$	D	$	D	d
05	E	%	E	%	E	%	E	e
06	F	&	F	&	F	&	F	f
07	G	'	G	'	G	'	G	g
08	H	(	H	(	H	(	H	h
09	I	)	I	)	I	)	I	i
0A	J	*	J	*	J	*	J	j
0B	K	+	K	+	K	+	K	k
0C	L	,	L	,	L	,	L	l
0D	M	—	M	—	M	—	M	m
0E	N	.	N	.	N	.	N	n
0F	O	/	O	/	O	/	O	o

（续）

	反向显示		闪烁显示		正向显示		正向显示	
	大写字母	特殊字符	大写字母	特殊字符	大写字母	特殊字符	大写字母	小写字母
	\$00－\$1F	\$20－\$3F	\$40－\$5F	\$60－\$7F	\$80－\$9F	\$A0－\$BF	\$C0－\$DF	\$E0－\$FF
10	P	φ	P	φ	P	φ	P	p
11	Q	1	Q	1	Q	1	Q	q
12	R	2	R	2	R	2	R	r
13	S	3	S	3	S	3	S	s
14	T	4	T	4	T	4	T	t
15	U	5	U	5	U	5	U	u
16	V	6	V	6	V	6	V	v
17	W	7	W	7	W	7	W	w
18	X	8	X	8	X	8	X	x
19	Y	9	Y	9	Y	9	Y	y
1A	Z	:	Z	:	Z	:	Z	z
1B	[	;	[	;	[	;	[	{
1C	＼	<	＼	<	＼	<	＼	|
1D	]	=	]	=	]	=	]	}
1E	∧	>	∧	>	∧	>	∧	～
1F	—	?	—	?	—	?	—	?

附录D　区位码、国标码、内码对照表

区位码	国标码	异形国标码	学习机内码		区位码	国标码	异形国标码	学习机内码	
10进制	16进制	16进制	16进制	10进制	10进制	16进制	16进制	16进制	10进制
1	21	A1	1D	29	22	36	B6	34	52
2	22	A2	1E	30	23	37	B7	35	53
3	23	A3	1F	31	24	38	B8	36	54
4	24	A4	20	32	25	39	B9	37	55
5	25	A5	21	33	26	3A	BA	38	56
6	26	A6	23	35	27	3B	BB	39	57
7	27	A7	24	36	28	3C	BC	3B	59
8	28	A8	25	37	29	3D	BD	3C	60
9	29	A9	26	38	30	3E	BE	3D	61
10	2A	AA	27	39	31	3F	BF	3E	62
11	2B	AB	23	40	32	40	C0	3F	63
12	2C	AC	29	41	33	41	C1	40	64
13	2D	AD	2A	42	34	42	C2	41	65
14	2E	AE	2B	43	35	43	C3	42	66
15	2F	AF	2D	45	36	44	C4	43	67
16	30	B0	2E	46	37	45	C5	44	68
17	31	B1	2F	47	38	46	C6	45	69
18	32	B2	30	48	39	47	C7	46	70
19	33	B3	31	49	40	48	C8	47	71
20	34	B4	32	50	41	49	C9	48	72
21	35	B5	33	51	42	4A	CA	49	73

（续）

区位码	国标码	异形国标码	学习机内码		区位码	国标码	异形国标码	学习机内码	
10进制	16进制	16进制	16进制	10进制	10进制	16进制	16进制	16进制	10进制
43	4B	CB	4A	74	64	60	E0	5F	95
44	4C	CC	4B	75	65	61	E1	60	96
45	4D	CD	4C	76	66	62	E2	61	97
46	4E	CE	4D	77	67	63	E3	62	98
47	4F	CE	4E	78	68	64	E4	63	99
48	50	D0	4F	79	69	65	E5	64	100
49	51	D1	50	80	70	66	E6	65	101
50	52	D2	51	81	71	67	E7	66	102
51	53	D3	52	82	72	68	E8	67	103
52	54	D4	53	83	73	69	E9	68	104
53	55	D5	54	84	74	6A	EA	69	105
54	56	D6	55	85	75	6B	EB	6A	106
55	57	D7	56	86	76	6C	EC	6B	107
56	58	D8	57	87	77	6D	ED	6C	108
57	59	D9	58	88	78	6E	EE	6D	109
58	5A	DA	59	89	79	6F	EF	6E	110
59	5B	DB	5A	90	80	70	F0	9F	111
60	5C	DC	5B	91	81	71	F1	70	112
61	5D	DD	5C	92	82	72	F2	71	113
62	5E	DE	5D	93	83	73	F3	72	114
63	5F	DF	5E	94	84	74	F4	73	115

（续）

区位码	国标码	异形国标码	学习机内码		区位码	国标码	异形国标码	学习机内码	
10进制	16进制	16进制	16进制	10进制	10进制	16进制	16进制	16进制	10进制
85	75	F5	74	116	90	7A	FA	79	121
86	76	F6	75	117	91	7B	FB	7A	122
87	77	F7	76	118	92	7C	FC	7B	123
88	78	F8	77	119	93	7D	FD	7C	124
89	79	F9	78	120	94	7E	FE	7D	125

封面设计：阎欢玲

ISBN 7-5053-0627-8/TP·99　定价：1.90元

中华学习机使用入门

电子工业出版社